



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií



# SOFTWAREVÉ VYBAVENÍ ŘÍDICÍHO SYSTÉMU STŘÍDAVÉHO DYNAMOMENTRU

## Diplomová práce

*Studijní program:* N2612 – Elektrotechnika a informatika

*Studijní obor:* 3906T001 – Mechatronika

*Autor práce:* **Bc. Pavel Kněbort**

*Vedoucí práce:* Ing. David Lindr, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# SOFTWARE FOR CONTROL SYSTEM OF AC DYNAMOMETER

**Diploma thesis**

*Study programme:* N2612 – Electrical Engineering and Informatics

*Study branch:* 3906T001 – Mechatronics

*Author:* **Bc. Pavel Kněbort**

*Supervisor:* Ing. David Lindr, Ph.D.



## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Pavel Kněbort**  
Osobní číslo: **M11000272**  
Studijní program: **N2612 Elektrotechnika a informatika**  
Studijní obor: **Mechatronika**  
Název tématu: **Softwarové vybavení řídicího systému střídavého dynamometru**  
Zadávací katedra: **Ústav mechatroniky a technické informatiky**

### Z á s a d y   p r o   v y p r a c o v á n í :

1. Seznamte se s technickým vybavením inovovaného laboratorního dynamometru.
2. Na základě zadaných požadavků navrhnete vlastnosti a datové struktury software řídicího systému.
3. Navržený software realizujte a otestujte jeho funkcionalitu.



Rozsah grafických prací: dle potřeby dokumentace  
Rozsah pracovní zprávy: 40–50 stran  
Forma zpracování diplomové práce: tištěná/elektronická  
Seznam odborné literatury:


- [1] JOHN, Karl-Heinz; TIEGELKAMP, Michael. IEC 61131-3: programming industrial automation systems : Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids. 2nd ed. New York : Springer, 2010. 390 s. ISBN 978-3-642-12014-5
- [2] SIEMENS AG. Simotion Reference, dokumentace systému, Edition 08/2008
- [3] Technical Specification, PLCopen - Technical Committee2 - Task Force, Function blocks for motion control, Version 2.0

Vedoucí diplomové práce: Ing. David Lindr, Ph.D.  
Ústav mechatroniky a technické informatiky  
Konzultant diplomové práce: Ing. Martin Diblík, Ph.D.  
Ústav mechatroniky a technické informatiky  
Ostatní konzultanti: Ing. Leoš Beran, Ph.D.  
Ústav mechatroniky a technické informatiky

Datum zadání diplomové práce: 10. října 2013  
Termín odevzdání diplomové práce: 16. května 2014

  
prof. Ing. Václav Kopecký, CSc.  
děkan



  
doc. Ing. Milan Kolář, CSc.  
vedoucí ústavu

V Liberci dne 10. října 2013

## Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

## **Poděkování**

Chtěl bych poděkovat vedoucímu mé diplomové práce Ing. Davidu Lindrovi Ph.D. za vedení při vypracovávání práce a dále bych chtěl poděkovat konzultantovi mé diplomové práce Ing. Martinu Diblíkovi, Ph.D. za cenné rady při vypracování práce.

## **Anotace**

Diplomová práce řeší tvorbu softwaru pro obsluhu střídavého dynamometru. Dynamometr je vybaven pohonem od firmy Siemens a řídicím panelem od firmy B&R Automation. Spojení mezi dynamometrem a řídicím panelem je zajištěno pomocí sítě Ethernet a komunikačního protokolu TCP/IP.

Úvod práce popisuje technické vybavení inovovaného laboratorního dynamometru, jsou zde i uvedeny základní informace o PLC automatech a TCP/ IP komunikačním protokolu.

Následující část se část se zabývá návrhem vlastností a datových struktur softwaru, jsou zde uvedeny základní znalosti o programování PLC automatů. Nalezneme zde i podrobnější informace o ST programovacím jazyku, ve kterém byl software napsán.

Poslední část práce popisuje realizaci navrženého softwaru řídicího systému. Je zde popsán program Simatic Scout od firmy Siemens, ve kterém byl výsledný software realizován. Nakonec se testuje správná funkce softwaru a ověřuje se kompatibilita mezi vytvořeným softwarem a softwarem ovládacího panelu.

Klíčová slova: střídavý elektrický stroj, dynamometr, programový blok pro dynamometr, programy SIEMENS,SIMOTION SCOUT.

## **Anotation**

This thesis focuses on creating software AC dynamometer. The dynamometer is fitted by Siemens and the control panel from B & R Automation. The connection between the dynamometer and the control panel is provided via Ethernet and the communication protocol TCP / IP.

Key Introduction of the thesis describes an innovative technical equipment laboratory dynamometer. There are some basic information about the PLC and TCP / IP communication protocol.

The following section describes the design of the properties and data structures, software, here are the basic knowledge of PLC programming. Here one can find more detailed information about ST programming language in which the software was written.

The last part describes the software implementation of the proposed control system. There is described Simatic Scout program from Siemens, which was implemented resulting software. Finally, test the correct operation of the software and verifies the compatibility between the created software and software control panel.

Keywords:, AC electric machine, dynamometer, program block for dynamometer, programs of Siemens, SIMOTION, SIMOTION SCOUT



Úvod.....	9
2  Dynamometr .....	10
3  Programovatelné automaty Simotion D .....	12
3.1  Softwarové vybavení.....	13
3.2  Architektura systému SIMOTION .....	14
3.3  Systémové komponenty .....	15
3.4  Komunikace systému SIMOTION a SINAMICS.....	15
3.4.1 Komunikační služby.....	16
3.4.2 Komunikační služby SIMATIC S7 .....	16
3.5  Multitasking .....	17
3.6  Způsoby programování .....	18
4  Komunikace se systémem SIMOTION .....	20
4.1  Protokol TCP/IP .....	20
4.2  Datová struktura odesílaných dat .....	20
4.3  Datová struktura přijatých dat.....	21
5  Vývojové prostředí SIMOTION SCOUT .....	23
5.1  Základní informace pro práci v SIMOTION SCOUT .....	23
5.1.1 Tvorba nového projektu .....	24
5.1.2 Archivace a obnovení projektu.....	25
5.1.3 Knihovny funkcí.....	25
5.1.4 Podpora multitaskingu.....	26
5.1.5 Seznam pro pokročilé uživatele .....	27
5.1.6 Měření .....	28
5.2  Technologické objekty .....	30
6  Návrh software pro střídavý dynamometr .....	31

6.1	Vlastnosti softwaru panelu 445 .....	31
6.1.1	Obslužné programy panelu.....	31
6.2	Vizualizace panelu 445 .....	32
6.3	Diagnostika chyb ovládacího panelu .....	35
6.4	Vlastnosti softwarových struktur .....	37
6.5	Strukturovaný text .....	37
7	Realizace software pro střídavý dynamometr .....	40
7.1	Proměnné.....	40
7.2	Struktura programování .....	40
7.3	Motion Control function blocks .....	41
7.4	Task skupiny .....	41
7.5	Komunikační vrstva .....	42
7.6	Hlavní programová vrstva.....	44
7.6.1	Usměrňovač.....	45
7.6.2	Řízení ventilátorů .....	45
7.6.3	Vyhodnocení chyb ovládacího panelu .....	46
7.7	Datová struktura dGlobal .....	46
7.8	Obslužná vrstva .....	47
8	Testování realizovaného softwaru pro úlohu dynamometru .....	52
	Závěr .....	57
	Použitá literatura .....	59

## Seznam použitých symbolů a zkratek

<b>Zkratka</b>	<b>Popis</b>	
<b>CPU</b>	<b>Central Processor Unit</b>	<b>Centrální procesorová jednotka</b>
<b>DCC</b>	<b>Driver control charts</b>	<b>Diagram pro řízení pohonu</b>
<b>EL</b>	<b>Expert list</b>	<b>Expert list</b>
<b>FB</b>	<b>Function block</b>	<b>Funkční blok</b>
<b>FBD</b>	<b>function block diagram</b>	<b>Diagram funkčních bloků</b>
<b>IL</b>	<b>instruction list</b>	<b>Seznam instrukcí</b>
<b>LTM</b>	<b>levé tlačítko myši</b>	
<b>MCC</b>	<b>Motion control chart</b>	<b>graf pro řízení pohybů</b>
<b>MCFB</b>	<b>Motion control function block</b>	<b>funkční blok pro řízení pohybů</b>
<b>PLC</b>	<b>Programmable Logic Controller</b>	<b>Programovatelný logický automat</b>
<b>POU</b>	<b>Programming organization unit</b>	<b>programová organizační jednotka</b>
<b>PTM</b>	<b>pravé tlačítko myši</b>	
<b>SSC</b>	<b>SIMOTION SCOUT</b>	
<b>ST</b>	<b>Structured text</b>	<b>strukturovaný text</b>
<b>TCP/IP</b>	<b>Transmission Control Protocol / Internet Protocol</b>	

## Úvod

Úkolem diplomové práce je vytvořit software pro systém Siemens SIMOTION dynamometru, ten se nachází v laboratořích Ústavu mechatroniky a technické informatiky a měl by v budoucnu sloužit k měření momentové charakteristiky motorů v laboratorních úlohách. Dynamometr je kromě systému SIMOTION vybaven Power Panelem 445 od firmy B&R Automation. Jde o PLC s integrovaným dotykovým displejem.

Cílem je vytvořit aplikaci, která bude kompatibilní s již existujícím softwarem pro panel 445.

Nejprve bylo nutné se seznámit s hardwarovým vybavením střídavého dynamometru, ten je vybaven následujícími prvky:

- Frekvenční měnič
- Synchronní servomotor
- Řídicí systém pro frekvenční měnič Siemens SINAMICS
- Externí měřicí systém

Podrobnější informace o programovatelných automatech SIMOTION jsou uvedeny v následující kapitole. Komunikace mezi panelem PP45 a systémem SIMOTION je uskutečněna v síti ethernet s využitím komunikačního protokolu TCP/IP.

Před samotnou realizací bylo nutné seznámit se blíže se systémem SIMOTION a na základě požadovaných vlastností tzn. s ohledem na již existující software pro panel 445 navrhnout programové vybavení.

Software pro systém SIMOTION byl vytvořen ve vývojovém prostředí SIMOTIN Scout. Zde bylo na výběr z několika programovacích jazyků, po zvážení všech možností byl zvolen strukturovaný text podrobnější informace lze nalézt v jedné z následujících kapitol. Pro lepší přehlednost byly programy rozděleny do tří vrstev.

Po realizování softwaru se provádí ověření jeho správné funkčnosti a také jeho kompatibility s řídicím softwarem panelu.

## 2 Dynamometr

Dynamometr je přístroj na měření krouticího momentu. Laboratorní dynamometr se skládá z několika částí:

- Externí měřicí systém
- Frekvenční
- Synchronní servomotor
- Řídicí systém SINAMICS
- Ovládací panel P445

Na následujícím obrázku je vyfocen laboratorní dynamometr

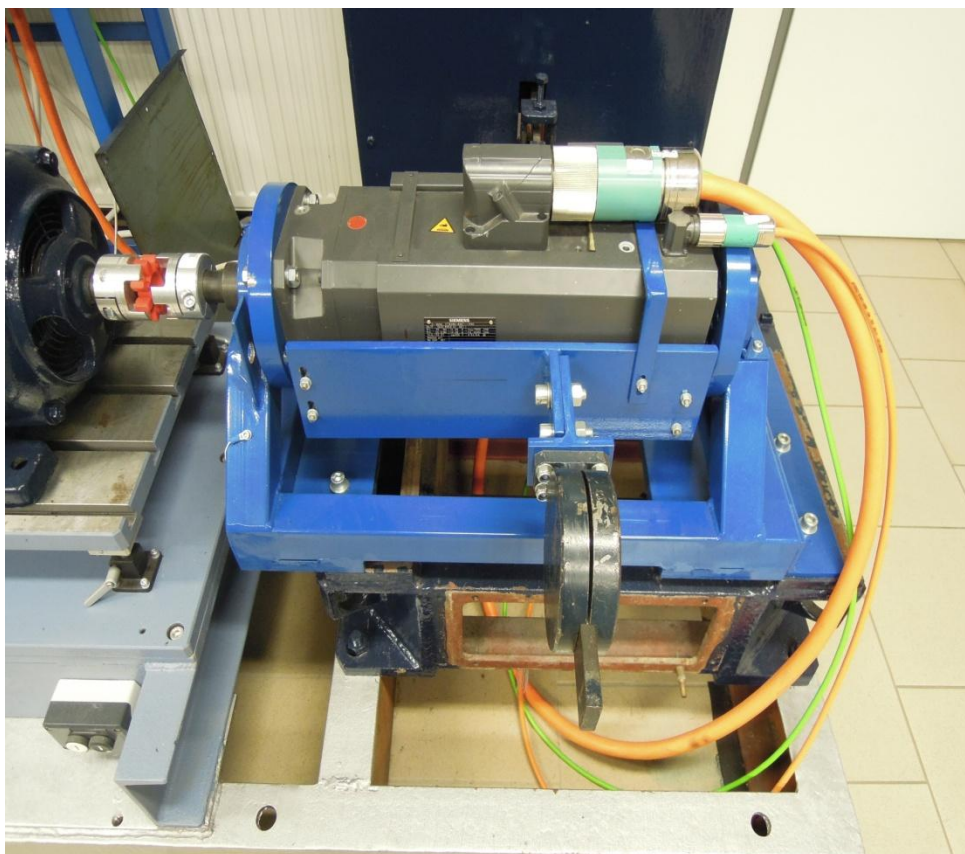


**Obr.2.1: Laboratorní dynamometr**

Dynamometr má tři druhy regulace:

- Rychlostní
- Polohová
- Momentová

Na Obr. 2.2 je vyfocen pohon dynamometru a na Obr. 2.3 externí měřicí systém.



**Obr.2.2: Pohon dynamometru**



**Obr.2.3: Externí měřící systém**

### **3 Programovatelné automaty Simotion D**

Kapitola byla zpracována pomocí [6] [7] [8].

Programovatelné logické automaty neboli PLC mají významné uplatnění především v průmyslu. Jsou využívány díky své vysoké spolehlivosti. Základním znakem všech PLC je, že pracují v cyklu, který se neustále opakuje.

PLC řady D mají kompaktní víceosé ovládání, které je vestavěno přímo do pohonu. Funkce SINAMICS S120 jsou přímo integrovány do řídicího modulu systému SIMOTION. Mezi hlavní výhody patří to, že celý systém se skládá ze dvou hlavních částí, ovládací části a hnací části, díky tomu je celý systém kompaktní.

Víceosý systém navíc nabízí velký stupeň přizpůsobivosti a velkou flexibilitu s ostatními jedno a víceosými řídicími systémy.

#### ***Použití programovatelných automatů***

Simotion řady D se hodí zejména pro aplikace, které mají následující parametry:

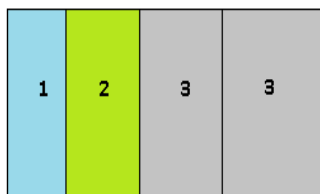
- Multiosé řízení (požaduje se řízení více os)
- Kompaktní stroje
- Časově kritické operace mezi vazbami
- Decentralizované automatizační koncepce, především pro stroje s větším počtem os.

#### ***Hardwarové komponenty***

SIMOTION D se skládá především z těchto komponent:

- Simotion D (řídicí jednotka)
- SINAMICS infeed (Line modul, usměrňovač)
- SINAMICS pohonná jednotka (motorové moduly)
- DRIVE-CLiQ

Podrobnější informace lze nalézt v [9].



1. SIMOTION D (Control unit), 2. SINAMICS Infeed (Line Module), 3. SINAMICS power units (Motor Modules)

### Obr. 3.1: Uspořádání SIMOTION D425

Na Obr. 3.2: je znázorněno klasické uspořádání SIMOTION D425.

## 3.1 Softwarové vybavení

Mezi základní softwarové vybavení patří SIMOTION systém a SINAMICS S120 zpětnovazební řízení motoru. SIMOTION systém obsahuje:

- Uživatelsky programovatelný runtime systém
- Komunikační funkce
- Funkce pro řízení pohybu
- Tasky
- Technologické objekty

SINAMICS S120 *drive control* zpřístupňuje následující funkce:

- proudové a momentové řízení
- rychlostní řízení
- infeed řízení

Rychlostní řízení se používá pro řízení otáček pohonu, momentové řízení se používá pro řízení pomocí velikosti momentu.



### **3.2 Architektura systému SIMOTION**

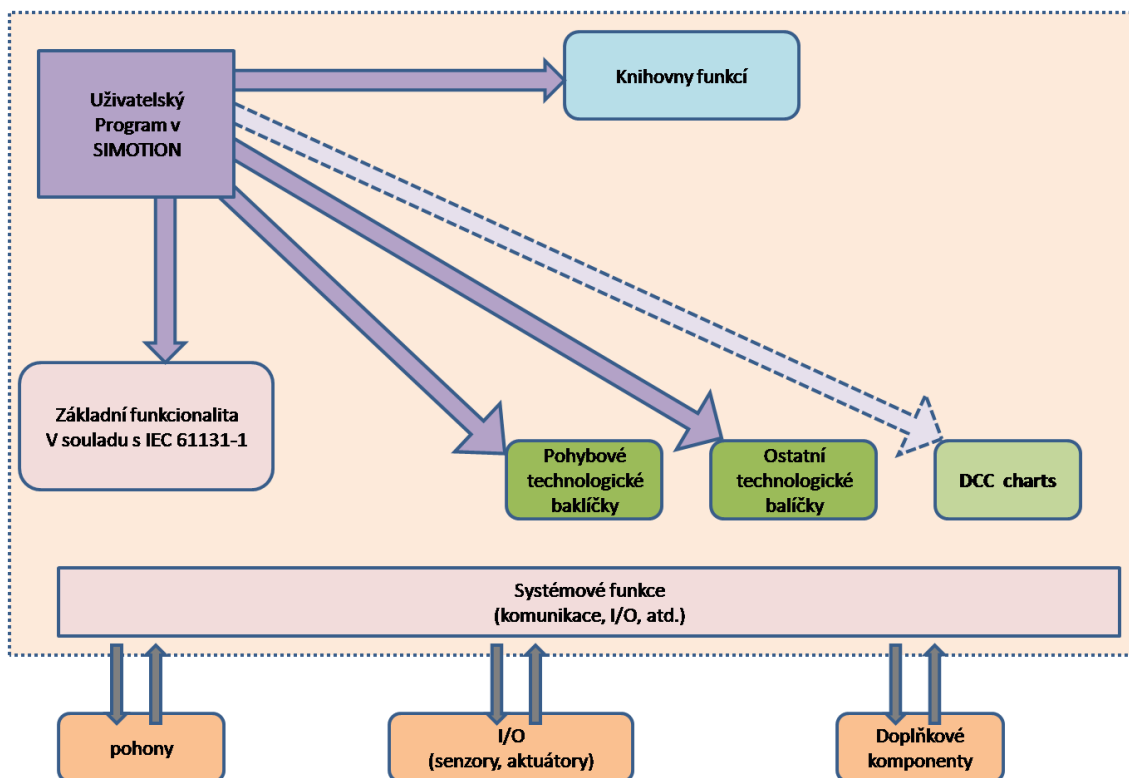
Architektura systému SIMOTION se může dále vyvíjet směrem k decentralizaci a distribuované inteligenci. Jak již bylo dříve uvedeno, obsahuje SIMOTION funkce pro open loop a closed loop řízení. Stejně tak obsahuje i aritmetické a logické funkce.

Provádění programu může být cyklické, časově spouštěné nebo vyvolané přerušením. SIMOTION kernel obsahuje funkce, které jsou dostačující pro většinu aplikací a v podstatě odpovídá PLC se sadou příkazů podle IEC 61131-3.

Možnosti SIMOTION Kernel můžou být rozšířeny pomocí přídavných technologických balíčků. Pro jednotlivé úlohy (tasky) můžou být použity již existující aplikace nebo v případě potřeby je může uživatel naprogramovat a propojit se zvolenými úlohami. Existující aplikace jsou naprogramovány v souladu s IEC 61131-3 a dají se přizpůsobit požadavkům jednotlivých úloh. Více informací o úlohách a jejich propojení viz 17

Navíc SIMOTION obsahuje knihovny, které obsahují systémové funkce a pohybové funkce. Kromě toho obsahují přístup k systémovým proměnným technologických objektů, které jsou napojeny na související technologické balíčky a zařízení v SScout.

Pro speciální úlohy jako je například zpětnovazební řízení mohou být použity grafická schémata, a vykonávat úlohy v DCC pomocí grafických bloků. Na Obr. 3. je zobrazena architektura systému SIMOTION.



Obr. 3.2: Architektura systému SIMOTION

### 3.3 Systémové komponenty

Podrobnější informace jsou uvedeny v [8]. SIMOTION D425 umožňuje komunikovat s automatizovanými komponenty přes rozhraní ethernet, PROFIBUS, PROFINET, DRIVE CLiQ.

### 3.4 Komunikace systému SIMOTION a SINAMICS

Komunikace mezi SINAMIC a SIMOTION probíhá pomocí PROFIBUS . Ten je standardizovaný v souladu s IEC 61158 a EN 50170. Používá se pro připojení periférií např. distribuovaných I/O, ventilů apod. je použit u automatizačních systémů jako Simotion, SIMATIC S7. Kromě jiného lze PROFIBUS využít i v prostorách, kde hrozí nebezpečí výbuchu.

#### ***Vlastnosti sběrnice PROFIBUS***

Oproti běžnému I/O disponuje PROFIBUS několika výhodami:

- Vestavěné diagnostické zařízení
- Nižší náklady na instalaci a zprovoznění
- Snadnější plánování

Existuje více typů provedení PB, posle toho pro jakou aplikaci je určen. Například PROFIBUS DP je určen pro rychlou komunikaci s inteligentními zařízeními distribuovaných I/O .

Profibus PA vede signály a napájení pro senzory a akční členy přes jednu linku.

Díky svému modulárnímu provedení a jednotnému komunikačnímu protokolu je sběrnice vhodná jak pro průmyslovou automatizaci tak i zpracovatelský průmysl.

Mezi další rysy patří zejména:

- Časová synchronizace
- Volitelná redundance pro hostitelský systém nebo zařízení typu slave
- Časové razítko

### **3.4.1 Komunikační služby**

Komunikační služby PQ/OP podporují protokol, který využívají kontrolory S7 pro komunikaci se zařízeními HMI popřípadě jinými programovatelnými zařízeními např. PC. Mezi typické zástupce HMI zařízení patří:

- Dotykové panely
- Ovládací panely
- Textová pole

Díky tomu, že jsou komunikační funkce S7 integrovány do systému SIMATIC, je možné přistupovat k datům v kontroloru, přes zařízení typu HMI, PG nebo PC.

Služby PQ/OP zpřístupňují následující funkce:

#### **STEP 7**

- Stažení konfigurace hardwaru
- Stažení uživatelských programů
- Online monitorování S7 (testování a diagnostika)

### **3.4.2 Komunikační služby SIMATIC S7**

Všechny kontrolory SIMATIC S7 a C7 umožňují uživatelskému programu číst a zapisovat data pomocí integrovaných komunikačních služeb. S7-400 používá SFBs a S300 popřípadě C7 využívají FBs.

Obě funkce jsou nezávislé na komunikační službě, která dovoluje S7 využívat komunikaci přes PROFIBUS nebo průmyslový ethernet.

Komunikační služby S7 poskytují především následující funkce:

- Během konfigurování systému je možné nakonfigurovat připojení, které mají být použita pro komunikaci S7.
- K jednomu partnerovi lze vytvořit i více připojení. Počet komunikačních partnerů kteří jsou neustále k dispozici je omezen CPU.
- Maximální velikost datového bloku je 64KB na jeden call SFBs nebo FB.

### **3.5 Multitasking**

Systém SIMOTION využívá tzv, multitaskingu. Tato podkapitola popisuje, jakým způsobem se zpracovávají jednotlivé úkoly. Systém umožňuje jednotlivým programům (taskům) přiřadit rozdílné úrovně (priority) spuštění. Ke každé úrovni spuštění jsou přiřazeny uživatelské programy. Každé úkoly jsou přiřazeny programům.

Systém používá následující úrovně třídy:

- Časově řízené úrovně
- Událostmi řízené úrovně
- Ovládané přerušením
- Synchronní
- Volně běžící
- Systémové úlohy (jsou pravidelně prováděny systémem)

Mezi úlohy, které jsou řízené systémem, patří komunikace a řízení pohybu.

#### **Spouštěcí úrovně**

Spouštěcí úrovně určují, v jakém pořadí systém jednotlivé úkoly vykoná. V každé úrovni se většinou nachází větší množství úkolů. Task poskytuje rámec pro spouštění jednotlivých programů. Úloha je provedena vždy, jestliže jsou splněny určité podmínky. Ke každé úloze může být přiřazeno více uživatelských programů. Jak již bylo uvedeno, kromě uživatelských programů existují i systémové úlohy. U systémových úloh nelze ovlivnit ani jejich obsah, ani posloupnost, v jaké se vykonávají.

#### **Úlohy**

Pro vlastní programování je k dispozici na každou prováděcí úroveň jedna, nebo více úloh. Mezi hlavní vlastnosti úloh patří to, za jakých podmínek je daná úloha

spuštěna a s jakou běží prioritou. Úloha může být přerušena jinou úlohou s vyšší prioritou. Následující úlohy jsou dostupné pro uživatelské programy.

- Úlohy po spuštění
- Volně běžící úlohy
  - Pohybové úlohy
  - Úlohy na pozadí
- Časově řízené a synchronní úlohy
  - Časově přerušované úlohy
  - Synchronní úlohy
- Událostmi řízené úlohy
  - Úlohy přerušované systémem
  - Úlohy přerušení uživatelem
- Vypínací úlohy

### **3.6 Způsoby programování**

Jak již bylo řečeno PLC automaty pracují v cyklu. Vývojové prostředí SIMOTION Scout umožňuje programovat v těchto programovacích jazycích:

- Function block diagram (FBD)
- Motion Control Chart (MCC)
- Drive Control Chart (DCC)
- Ladder diagram (LD)
- Structured text (ST)

Všechny výše uvedené programovací jazyky můžeme rozdělit do dvou skupin, na vyšší a nižší programovací jazyky. Dalším zásadním kritériem je způsob tvorby programu. Kromě standardních programovacích jazyků, kde se kód tvoří použitím textových příkazů, zde existují i tzv. grafické programovací jazyky.

Mezi zástupce grafických programovacích jazyků patří například programování pomocí FBD. Zde je celý program tvořen pomocí grafických bloků, které představují jednotlivé funkce, operátory atd. FBD je vhodné pro tvorbu jednodušších aplikací, díky grafické reprezentaci je tvorba programu velmi intuitivní a není tak časově náročná oproti programování např. ve strukturovaném textu (ST). Pokud jsou však za potřebí komplexnější programové struktury, je vhodné použít programovací jazyk z vyšší mírou abstrakce. [2].

### ***Výběr programovacího jazyka***

V kapitole 3.6 jsou uvedeny nejvýznamnější způsoby programovacích jazyků pro PLC Siemens. Po zvážení všech možností byl jako programovací jazyk zvolen strukturovaný text. Nižší programovací jazyky nebyly zvoleny, protože neposkytují tak přehlednou konstrukci programu (např. FB). Bližší popis vybraného programovacího jazyka lze nalézt v kapitole 6.5

## 4 Komunikace se systémem SIMOTION

Panel 445 je se systémem SIMOTION propojen pomocí sítě ethernet. Ke komunikaci používá protokol tcp/ip. Komunikace mezi Siemens Sinamics a B a R řídicím systémem je zajištěna pomocí tzv. datových balíků [1]

### 4.1 Protokol TCP/IP

Je velmi často používán např. internet. Rodina TCP/IP protokolů obsahuje čtyři vrstvy. V transportní vrstvě se nacházejí protokoly TCP a UDP. Protokoly slouží k zajištění spojení mezi aplikacemi, které běží na vzdálených počítačích.

K přenosu dat TCP používá datové pakety. TCP mezi dvěma aplikacemi naváže spojení tak, že vytvoří po dobu spojení virtuální okruh. Ten je plně duplexní. Podrobnější informace o TCP/IP komunikace lze nalézt v [5].

### 4.2 Datová struktura odesílaných dat

Jak bylo již uvedeno, komunikace probíhá v telegramech. Každý telegram se skládá z pole datových proměnných typu real o velikosti [ 0..9 ]. Následující tabulka uvádí strukturu dat, které jsou odesílány systémem SIMOTION.

**Tab. 4.1: Struktura odesílaných dat**

Index	Uložené hodnoty
0	Stavové slovo
1	Aktuální otáčky
2	Aktuální poloha
3	Aktuální moment
4	Aktuální akcelerace
5	Aktuální decelerace
6	Aktuální proud
7	Aktuální výkon
8	Aktuální teplota usměrňovače
9	Aktuální teplota motoru

Stavové slovo je klíčové pro řízení dynamometru, jsou v něm uvedeny všechny aktuální nastavení systému SIMOTION. Stavové slova slouží zároveň ke kontrole úspěšného nastavení systému dle obdrženého řídicího slova. Stavové slovo je v systému uloženo jako proměnná typu bool, před odesláním je tedy nutné převést typ bool na typ Real. Následující tabulka znázorňuje strukturu stavového slova.

**Tab. 4.2: Stavové slovo pro nastavení dynamometru**

Index bitu	Funkce	Nastavení	
0	Pohon v chodu	1	0
1	Pohon zastaven	1	0
2	Momentová regulace	1	0
3	Otáčková regulace	1	0
4	Polohová regulace	1	0
5	Dynamometr charakter pohon	1	0
6	Dynamometr charakter brzda	1	0
7	Otáčky – vpravo	1	0
8	Otáčky – vlevo	1	0
9	Napájení usměrňovače	1	0
10	Ventilátor usměrňovače	1	0
11	Ventilátor motoru	1	0
12	Požadovaná hodnota byla dosažena	1	0
13	Akcelerace	1	0
14	Decelerace	1	0
15	Chyba	1	0

### **4.3 Datová struktura přijatých dat**

Systém SIMOTION přímá datové balíky od systému B & R jejich struktura je popsána v tabulce Tab. 4.3.



**Tab. 4.3: Struktura přijatých dat**

Index REALu	Uložené hodnoty
0	Řídící slovo
1	Otáčky /Moment/poloha
2	Akcelerace
3	Decelerace
4	Momentová rampa
5	Volno
6	Volno
7	Volno
8	Volno
9	Volno

Řídící slovo slouží systému Siemens SIMOTION k nastavení všech potřebných parametrů, které jsou nutné pro provoz dynamometru jak je patrné z předešlé tabulky je řídící slovo v REALu , aby bylo možné použít přijaté řídící slovo, musí být z typu REAL převedeno na typ BOOL. Tabulka č.4 ukazuje struktura řídícího slova používaného systémem Siemens.

**Tab. 4.4: Struktura přijatého řídícího slova**

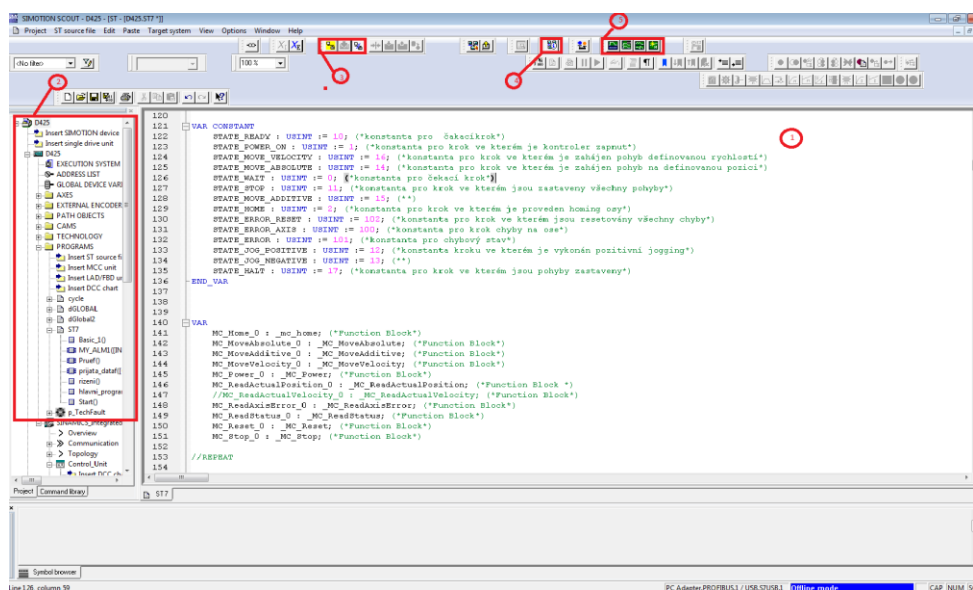
Index bitu	Funkce	Nastavení	
0	Osa motoru	1	0
1	Momentová regulace	1	0
2	Otáčková regulace	1	0
3	Polohová regulace	1	0
4	Charakter dynamometru	Pohon	Brzda
		1	0
5	Směr otáčení	vpravo	vlevo
		1	0
6	Napájení usměrňovače	Zapnuto	Vypnuto
		1	0
7	Ventilátor usměrňovače	Zapnuto	Vypnuto
		1	0
8	Ventilátor motoru	Zapnuto	Vypnuto
		1	0
9	ERROR Potvrzení	Ano	Ne
		1	0
10-15	Volno		

## 5 Vývojové prostředí SIMOTION SCOUT

Jde o vývojové prostředí, sloužící k programování řídicího systému SIMOTION dodávané firmou siemens. Systém SIMOTION pracuje na principu multitaskingu, který bude později podrobněji popsán. Kapitola se zabývá základním rozhraním programu, jsou zde uvedeny informace o technologických objektech. Nalezneme zde i krátký úvod do měření pomocí SSC. Kapitola byla zpracována s využitím [10].

### 5.1 Základní informace pro práci v SIMOTION SCOUT

V této podkapitole je uvedeno několik základních informací o práci v SIMOTION Scout. Především jsou zde popsány praktické informace o vytváření nového projektu, připojení se k cílovému zařízení apod. Na následujícím obrázku je popsáno základní uživatelské rozhraní.



Obr. 5.1: Rozhraní programu Siemens SIMOTION SCOUT

Popisky: 1. Pracovní plocha 2. Stromový diagram projektu 3. Ikony pro přechod do online stavu, stažení programu do zařízení a odpojení zařízení (offline režim) 4. Accesible nodes 5. Nástroje pro měření

### **5.1.1 Tvorba nového projektu**

Založení nového projektu je velmi intuitivní, při tvorbě projektu zadáme informace o použitém pohonu (výkon, typ, počet os).

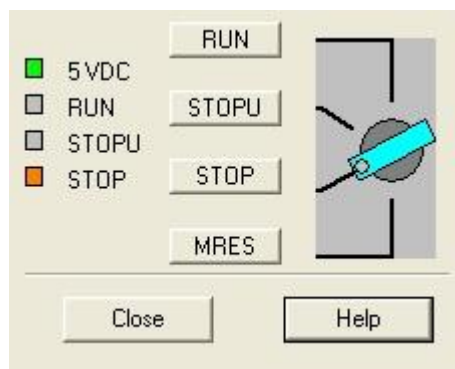
#### ***Konfigurace komunikace***

Při konfiguraci systému je vhodné nastavit komunikaci. Jak již bylo dříve uvedeno komunikace mezi ovládacím panelem a SIMOTION je realizována pomocí ethernetu. Komunikace mezi SIMOTION a systémem SINAMICS probíhá pomocí sběrnice PROFIBUS viz 3.4

#### ***Práce v online stavu***

K tomu aby bylo možné vlastní programy vyzkoušet, musejí být nejdříve nahrány do cílového zařízení. Při zapnutí projektu nejsme připojeni k hardwaru, tzn. nacházíme se v offline stavu, k přechodu do online stavu je nutné zvolit položku go to online viz Obr. 5.1 . Pokud nelze přejít do online stavu, může to mít mnoho příčin. Nejčastějším a snadno odstranitelným problémem je, že SScout nevyhledal všechna existující spojení. Klikneme LTM na položku *Accesible nodes* (4. Položka v Obr. 5.1) systém sám automaticky vyhledá všechny existující spojení a zobrazí je. V online stavu můžeme námi požadované programy nahrát do hardwaru a následně je spustit. Vývojové prostředí nás samo pomocí grafických prvků informuje, jsou li programy v projektu shodné s programy nahrané v hardwaru. Díky tomu máme vždy přehled, je li nahraná verze programu aktuální. Jestliže jsou verze shodné, je vedle ikon zařízení zobrazena zelená ikona, pokud jsou verze rozdílné, barva se změní na zelená-červená. Pokud se nelze připojit k určitému zařízení je ikona červená.

I přestože vývojové prostředí SIMOTION SCOUT obsahuje vlastní nástroj pro testování programů, jsou jeho funkce dosti omezené a není vhodné ho využívat pro testování komplexnějších programů. Pokud jsme úspěšně přešli do online stavu a úspěšně nahráli program do zařízení, je dobré zkontrolovat, zda je modul ve stavu *RUN*. Pokud by tomu tak nebylo program je sice v zařízení správně nahrán, ale neběží, proto by se nepodařilo zapnout usměrňovač pomocí dotykového panelu. Stav zařízení lze zkontrolovat pomocí signalizace diod. Na obrázku je zobrazeno ovládání operačního módu zařízení D425.



**Obr. 5.2: Panel pro aktuální stav PLC D425**

### ***Okno s alarmy***

V okně *alarms* jsou zobrazena všechna upozornění, které se týkají všech chyb na zařízení. Jestliže uživatel nepotvrdí pomocí tlačítka *acknowledge* nebo *acknowledge all*, že si je vědom dané chyby, zařízení zůstane v chybovém stavu.

### ***5.1.2 Archivace a obnovení projektu***

Velmi užitečným nástrojem je archivace a obnovení projektu. Díky této funkci můžeme celý projekt zálohovat do jediného rar souboru, to je velmi užitečné pokud potřebujeme projekt přenést do jiného počítače. K funkci archivace se dostaneme přes nabídku projekt, a zde z menu vybereme *archive*. Potřebujeme-li projekt obnovit, postupuje podobně, ale místo archivace zvolíme položku obnovení s archivu. Pokud byl program vytvořen ve starší verzi SSC, než která je použita pro obnovení z archivu, pravděpodobně bude nutné provést konverzi do novější verze. Konverze probíhá automaticky, ale je nutné si uvědomit, že projekt převedený do novější verze nebude moci ve starších verzích otevřít. Z tohoto důvodu je nutné každou konverzi potvrdit.

### ***5.1.3 Knihovny funkcí***

Programové prostředí umožňuje využívat mnoho funkcí, jak již bylo dříve uvedeno např. systémové funkce nebo pohybové funkce. Protože se jich zde nachází značné množství, jsou funkce rozděleny do několika skupin, podle toho k jakým účelům jsou funkce určeny. Toto třídění usnadňuje uživateli orientaci. Pro tuto práci byly využity především funkce z těchto skupin:

- Communication (funkce pro zahájení komunikace, posílání a přijímání dat, ukončení komunikace apod.)
- Drives (SINAMICS použita struktura FB ALM Control viz)

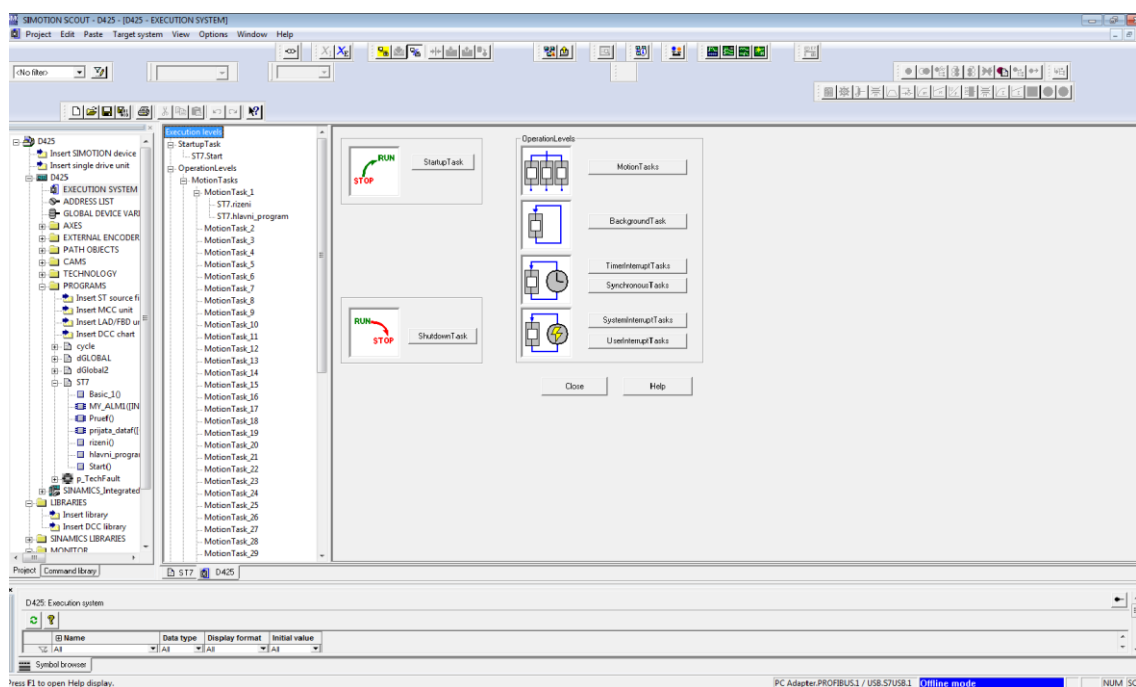
- PLC Open (Funkční bloky především pro jednoosé a víceosé operace, v této práci byli použity jen funkce určené pro jednu osu)
- System data -> Data information (obsahuje velmi užitečné funkce pro práci s daty např. funkce pro zjištění velikosti dat apod.)

Pokud by uživatel potřeboval vyhledat konkrétní funkci je k dispozici indexové vyhledávání.

#### **5.1.4 Podpora multitaskingu**

Jak již bylo uvedeno systém SIMOTION podporuje multitasking viz. Aplikace SIMOTION SCOUT umožňuje upravit pořadí, ve kterém budou jednotlivé programy vykonány, pomocí rozhraní *execute tasks system* . Postup k přístupu k rozhraní pro úpravu vykonávání úloh je následující: pravým tlačítkem myši klikneme na položku *PROGRAMS* která se nachází v levém okně viz Obr. 5.3 a z rozbaleného menu levým tlačítkem myši vybereme položku *Configure execution system*. Mnohem jednodušší způsob, jak přejít k rozhraní pro úpravu vykonávání úloh je vybrat položku *execute task systém*, která se nachází ve stromovém menu, hned pod D425.

Jestliže daný program není přiřazen k žádnému tasku, tak i přesto, že je nahrán do zařízení se neprovede. Jeden program může být přiřazen do více tasků. Počet do kolika skupin je program přiřazen se uvádí pomocí čísla v závorce, které se nachází vedle názvu programu. Tento systém vykonávání příkazů je velmi užitečný například při ladění programu, pokud potřebujeme, můžeme kterýkoliv z podprogramů dočasně vyřadit pomocí *execute task systemu*.



**Obr. 5.3: Rozhraní pro řízení vykonávání úloh (*exucute task system*).**

### 5.1.5 Seznam pro pokročilé uživatele

Expert List slouží pro snadnější vyhledání požadovaných systémových proměnných. Umožňuje přístup ke konfiguračním datům. Většina těchto dat je dostupná i přes průvodce a parametrizační formuláře, které jsou přehlednější.

Jedním z příkladů, kdy je použití expert listu vyžadováno, může být změna konfiguračních dat v online stavu. SIMOTION Scout od verze 4.1 poskytuje oddělené zobrazení pro EL. Nachází se zde vybraná konfigurační data a nejvýznamnější systémové proměnné (parametry). To je velmi užitečné zejména pro diagnostiku a programování, protože díky EL máme snadný přístup k těmto vybraným parametrům.

Pokud využijeme úpravy parametrů pomocí EL, měly bychom si uvědomit, že systém neprovádí kontrolu parametrů, které souvisejí s upravovanou hodnotou.

Následující obrázek zachycuje přehled parametrů. Jak je patrné z obrázku, parametry jsou barevně odlišeny. Pokud lze parametr upravovat je vybarven zeleně. Je-li možná úprava v offline režimu je vyznačen žlutě. Šedivě jsou vyznačeny parametry, které není možné upravovat.

Param.	Name	Unit	Modifiable to	Access level	Minimum	Maximum
14	Drive operating display	[42] [42] Power-on stb.	1	All		
2	BOP operating display selection, Parameter number	2	Operation	2	0	65535
3	BOP operating display mode	[4] 0000	Operation	3		
4	Drive commissioning parameter filter	[0] Ready	Ready to run	1		
5	BOP user-defined bit	0	Operation	3	0	65535
6	Macro drive object	0	Commission	1	0	999999
7	Speed setpoint, smoothed	0.0	RPV	2		
8	CO Actual speed, smoothed	-0.2	RPV	2		
9	Speed actual value rpm, smoothed	-0.2	RPV	2		
10	Output frequency, smoothed	0.0	Hz	3		
11	CO Output voltage, smoothed	0.0	Vrms	2		
12	CO DC link voltage, smoothed	1.2	V	2		
13	CO Absolute actual current, smoothed	0.00	Amps	2		
14	Modulation depth, smoothed	0.0	%	3		
15	Current actual value field-generating smoothed	-0.00	Amps	3		
16	Current actual value torque-generating smoothed	0.00	Amps	3		
17	Actual torque smoothed	0.00	Nm	2		
18	CO Active power actual value, smoothed	-0.00	kW	2		
19	Torque utilization, smoothed	0.0	%	3		
20	CO Motor temperature	19.0	°C	2		
21	Power unit overload I2t	0.0		3		
22	CO Power unit temperatures, Maximum inverter	29	°C	3		

Obr.5.4 :Expert list

### 5.1.6 Měření

Jednou z důležitých funkcí programu Sscout je měření hodnot např. rychlosti, zrychlení, momentu atd. Následující podkapitola tvoří pouze základní úvod do měření a podrobnější informace lze nalézt v [10].

Pro samotné měření lze vybírat ze dvou systémů pro měření, buď je měření prováděno pomocí SIMOTION, nebo měření probíhá v SINAMICSU. Měření pomocí SINAMICSU je vhodné především pro:

- Měření rychlých dějů
- Měření dynamických dějů pohonu

Naopak měření v SIMOTION má následující přednosti:

- Lze nahrávat i dlouhodobé měření
- Velké množství měřících kanálů

Nástroje pro měření lze zobrazit kliknutím LTM na ikonu measurement. Mezi hlavní výhody měření v Sc patří:

- Sledování více měřených hodnot současně
- Nastavení podmínek pro zapnutí, vypnutí měření
- Značné množství parametrů, které lze sledovat
- Barevné odlišení sledovaných parametrů

Doba měření, kterou lze zaznamenat, je omezena především velikostí paměti. Toto omezení se vztahuje na měření využívající systém SINAMICS, protože oproti SINAMICSU jsme omezeni velikostí vnitřní paměti. Chceme-li zaznamenat větší množství parametrů současně, zkrátí se nám maximální doba měření. Dalším důležitým faktorem, který má vliv na maximální délku měření je vzorkovací frekvence, tedy kolikrát za sekundu má provést a zaznamenat měřená vybraných hodnot. Obrázek zobrazuje rozhraní pro měření.

Další nabídka nástrojů pro měření je přístupná přes položku trace. Zde si sami můžeme vybrat z velké nabídky dostupných parametrů, které bychom chtěli sledovat.

### ***Nahrávání měření***

Při měření je kromě možnosti nahrávání více parametrů současně velmi podstatné i to aby bylo nahrávání včas zapnuto. Kromě okamžitého nahrávání () máme k dispozici trigry které udávají podmínky, za kterých se má nahrávání automaticky spustit. Někdy kromě samotného jevu, který chceme měřit, by bylo vhodné zachytit i určitý časový úsek před výskytem jevu. K tomu nám slouží prediktivní nahrávání, kdy je zachycen i časový úsek před spuštěním nahrávání pomocí triggeru.

### ***Watch table***

Při tvorbě programu nastanou situace, kdy je potřeba znát hodnotu vybraných parametrů, nebo proměnných v reálném čase. Protože není třeba tyto hodnoty zaznamenávat, není nutné použít měřicí funkce.

K zjištění aktuální hodnot parametrů, či proměnných lze v programu SScout použít tzv. watch table. Jak již název naznačuje, jde o tabulku, kde mohou být zobrazeny námi sledované proměnné. Protože při složitých programech se může vyskytnout velké množství parametrů a proměnných, které je nutné sledovat, je vhodné vytvořit pro sledování více watch tabelů. Na obr. je zachycena tabulka pro sledování proměnných. Následující odstavec uvádí způsob jak vložit do SScout novou tabulku a přidat do ní požadované parametry.

Pro vložení nové tabulky pro sledování (watch table) otevřeme v projektovém navigátoru složku monitor. Dvakrát klikneme na položku *insert watch table*. Novou tabulku libovolně pojmenujeme. Vložení parametru či proměnné lze provést více způsoby. Můžeme využít například expert listu viz kapitola 5.1.5. Poté co požadovaný



parametr zkopírujeme z EL, klikneme PTM na prázdné pole v tabulce a dáme vložit (paste). Kromě expert listu lze k vyhledání parametrů použít i projektový navigátor. Siemens Simotion Scout verze 4.3 obsahuje rozšířené rozhraní pro správu watch table oproti předchozím verzím je přehlednější a usnadňuje uživateli vkládání nových parametrů.

## **5.2 Technologické objekty**

Technologické objekty (TO) poskytují funkce pro řízení pohybu a dodatečné technologie. Kromě toho umožňují uživateli lépe se zorientovat v technologii pohonů a senzorů. Technologické objekty lze využít pro značné množství aplikací např:

- TO osy (TO axis) pro pohon a enkodér
- TO cam pro využití složitých programových funkcí
- TO externího enkoderu (TO external encoder) pro jedno externí čidlo

V systému SIMOTION jsou TO dostupné pomocí technologických balíčků (technology packages), které jsou spolu vloženy do SIMOTION run time systému spolu s projektem.

Následující technologické balíčky jsou podporovány v systému SIMOTION:

- TP Cam obsahuje základní TO například pro řízení pohybu, mezi něž patří driveAxis, followingAxis apod.
- TP Cam\_ext
- TControl obsahuje technologické objekty pro teplotní regulátor
- TP Path

Aktivace nebo deaktivace technologických funkcí se provádí pomocí speciálních příkazů. Příkazy lze rozdělit do dvou skupin na synchronní a asynchronní. Synchronní příkazy pro pohyb je vhodné použít například při programování pohybové sekvence v sekvenčních úlohách (motion tasks).

Jednotlivé příkazy jsou přiřazeny k jednotlivým úlohám, které jsou řazeny v tzv. spouštěcích úrovních (execute task level) viz. 3.5. Příkazy lze vykonávat ze všech programových úloh. Protože doba provedení příkazu na technologickém objektu je jediným faktorem, který určuje rychlost provedení příkazu, a jestliže uživatel vydává příkazy z více různých úloh, musí být programová konzistence zajištěna uživatelským programem.

## 6 Návrh software pro střídavý dynamometr

Jak bylo uvedeno již dříve, výsledný software musí být schopen pracovat s již realizovaným softwarem panelu 445, proto museli být zohledněny i jeho vlastnosti.

Podkapitoly 5.1 až 5.3 jsou zaměřeny na popis dotykového panelu a jeho softwaru. Kapitola byla zpracována s využitím [1].

### 6.1 Vlastnosti softwaru panelu 445

Tato podkapitola vychází z [1], jsou zde uvedeny vlastnosti softwaru dotykového ovládacího panelu, které mají vliv na strukturu realizovaného softwaru pro řídicí systém. Software panelu odesílá řídicí slovo jako typ *real* přesto, že původně bylo typu *bool*. Pro systém SIMOTION by bylo samozřejmě výhodnější, pokud by řídicí slovo obdržel přímo, jako typ *bool*, ale vzhledem ke zvolené struktuře komunikace to není možné. Jak autor uvádí, komunikaci pomocí datových balíčků typu *REAL*, zvolil kvůli přesné interpretaci odeslaných dat. Je také pravda, že zadání rychlosti, akcelerace nebo decelerace se zadávají jako *LREAL*.

#### 6.1.1 Obslužné programy panelu

Aplikace řídicího panelu obsahuje tři obslužné programy. Jak autor uvádí hlavním důvodem je, aby došlo k rozdělení tří hlavních funkcí aplikace do tří programů. Programy pracují nezávisle na sobě. Mezi hlavní funkce aplikace patří:

- Řízení chodu dynamometru
- Kontrola chyb vzniklých v době provozu dynamometru.
- vizualizace

O řízení chodu dynamometru se stará program *Controls*, Pro vytvoření aplikace pro systém SIMOTION je důležitá především jeho inicializační část, zejména pak nastavení hodnot při inicializaci programu. Nejvýznamnější hodnoty jsou uvedeny v tabulce č.6, podrobnější informace o celém programu jsou uvedeny v [1].

Tab. 6.1: Mezní hodnoty nastavení ovládacího panelu

Popis proměnné	Nastavení:
Spouštění pohonu	0
Minimální hodnota otáček	0 ot./min
Maximální hodnota otáček	7200 ot./min

Minimální hodnota momentu	0 Nm
Maximální hodnota moment	35 Nm
Minimální hodnota akcelerace	0 ot./min/s
Maximální hodnota akcelerace	500 ot./min/s
Minimální hodnota decelerace	0 ot./min/s
Maximální hodnota decelerace	500 ot./min/s

Hodnoty uváděné v tabulce jsou mezní hodnoty, které lze na ovládacím panelu nastavit, je třeba si uvědomit, že pokud by maximální hodnoty, které lze na panelu zadat přesáhly maximální limity pohonu systém SIMOTION automaticky detekuje, že zadaná hodnota otáček je mimo bezpečný rozsah a zastaví pohon spolu s příslušným chybovým hlášením.

## 6.2 Vizualizace panelu 445

Protože mnoho vlastností je určeno vizualizací panelu uvedu zde základní informace o vizualizaci, podrobnější informace jsou uvedeny v [1]. Pro řízení dynamometru slouží uživateli barevný dotykový displej s rozlišením QVGA 320×240 úhlopříčkou 5,7“, samotné řízení je prováděno přes tzv. vizualizaci. Vizualizace má následující části (obrazovky):

- Nastavení pro provoz dynamometru
- Ovládání dynamometru spolu se zobrazením stavu aktuálně sledovaných veličin
- Zobrazení hodnot otáček a momentu v čase
- Nastavení systému
- Deník chyb

### Setup

Obrazovka *setup* umožňuje snadnou editaci systémových nastavení jako je systémový čas, jazyk rozhraní nebo nastavení sítě ethernet. Jak je uvedeno v [1] je systém dynamometru v pozici serveru a ovládací panel má pozici klienta. Pro správnou funkci komunikace je proto nutné správné nastavení ip adresy serveru (systém dynamometru) a zvolit číslo portu na kterém bude komunikace v síti ethernet probíhat.



**Obr. 6.1: Obrazovka setup**

### ***Nastavení***

Než bude možné pohon úspěšně provozovat, je nutné ho nejdříve nastavit. Nastavení se provádí pomocí dotykového panelu přes obrazovku nastavení. Zde je nutné zvolit požadovaný typ regulace, následuje volba směru otáčení pohonu. Uživatel může též nastavit způsob měření, a to buď manuální anebo automatický. Obrazovka nastavení obsahuje i rozhraní pro zapnutí napětí usměrňovače a ovládání větráku pohonu a usměrňovače. Na Obr. 6.2 je zobrazena obrazovka nastavení.

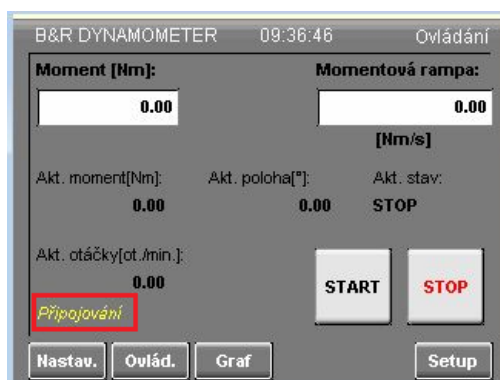


**Obr. 6.2: nastavení dynamometru pomocí dotykového panelu**

### ***Ovládání***

Na obrazovce ovládání probíhá nastavení hodnot regulované veličiny (otáčky, moment nebo poloha). Uživatel zde také nastaví požadovanou hodnotu akcelerace, decelerace nebo momentové rampy. V pravé dolní části jsou tlačítka pro zapnutí a vypnutí pohonu. Nastavení všech požadovaných číselných hodnot se provádí pomocí tzv. *NUMPadu*, po kliknutí na políčko s požadovanou veličinou se zobrazí klávesnice, pomocí které uživatel zadá požadovanou hodnotu. Zadaná hodnota se potvrdí tlačítkem *ENTER*.

Jestliže zadaná hodnota překračuje limit pro bezpečný provoz dynamometru, je tato hodnota nahrazena maximální bezpečnou hodnotou parametru. Díky této vlastnosti by nemělo dojít k poškození dynamometru zadáním příliš velké hodnoty například rychlosti nebo momentu. Na obrazovce ovládání jsou rovněž zobrazeny aktuální hodnoty momentu, otáček, polohy, kromě těchto údajů je zde i uveden aktuální stav pohonu. V levé dolní části je vidět stav komunikace viz. Obr. 6.3 Pokud jsou všechna požadovaná pole vyplněna lze pohon zapnout pomocí tlačítka START, po jeho stisknutí se zobrazí potvrzovací zpráva zda má být pohon zapnut, tato vrstva brání nechtěné aktivaci pohonu.



**Obr. 6.3: Obrazovka ovládání pohonu dynamometru**

### ***Blokace pohonu***

Jestliže by se během provozu pohonu vyskytla chyba, je nutné aby byl pohon nejen zastaven, ale i opětovné spuštění pohonu musí být blokováno do doby než je chyba odstraněna, nebo potvrzena. Po dobu trvání chyby je nad tlačítkem start zobrazen červený popisek chyby.

### ***Graf rychlosti pohonu***

Dotykový panel umožňuje vykreslit graf sledovaných veličin, jako jsou například rychlost nebo moment. Pro základní měření je toto grafické znázornění dostačující, ale pro měření více hodnot, popřípadě měření s více nástroji je vhodné použít buď měřicí nástroje SIMOTION, nebo SINAMICS viz. 5.1.6

### ***Chyby pohonu***

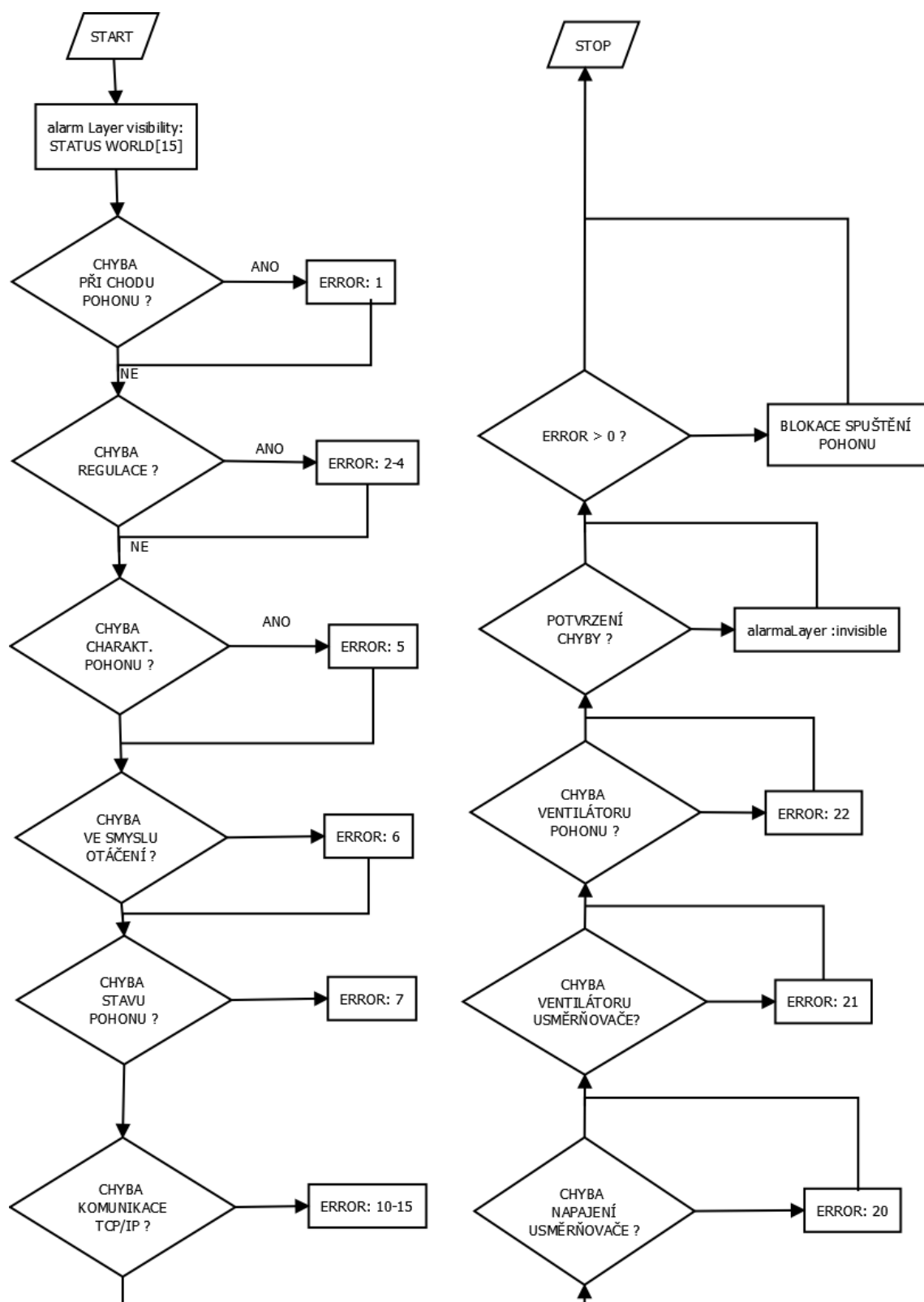
Vrstva se stará o výpis aktuálních chyb pohonu. Kromě výpisu aktuálních chyb lze zobrazit i historii chyb pomocí tlačítka historie.

### **6.3 Diagnostika chyb ovládacího panelu**

Aby bylo možné navrhnout software pro ovládání dynamometru bylo nezbytné zjistit, jakým způsobem dotykový panel detekuje chyby vzniklé na dynamometru a jak reaguje na tyto chyby. Jak je uvedeno v [1] program errors má v OP na starosti diagnostiku chyb. Z pohledu programování dynamometru jsou pro nás zásadní především následující funkce diagnostiky:

- Zjištění chyb TCP/IP komunikace mezi dotykovým ovládacím panelem a systémem Siemens SIMATIC
- Kontrola chybného nastavení systému SIMATIC
- Zajištění bezpečného provozu laboratorního dynamometru

Kontrola chyb proběhne v několika krocích první co je kontrolováno je zda se na pohonu nevyskytla chyba, dále je kontrolováno zda byla správně vybrána regulace, charakter pohonu (brzda motor), zde není zajištěno, že pokud bude vybrána regulace otáčková v generátorickém režimu nebude možné zapnout pohon dynamometru, pokud byly zvoleny nízké otáčky . Naopak je zde zajištěno kontrola stavu tcp/ip komunikace.



**Obr. 6.4: Průběh diagnostiky chyb ovládacího panelu**

Stavový diagram ukazuje postup programu ERRORS při diagnostice. Ve stavovém diagramu jsou uvedeny jednotlivé hodnoty proměnné ERROR. Její hodnota určuje index chyby.

### ***Potvrzení chyb***

Jak bylo již uvedeno, chyby pohonu se zobrazují ve vizuální vrstvě *Alamlayer*. Pokud nastane chyba pohonu, nebo usměrňovače stane se tato vrstva viditelnou a zobrazí se příslušné chybové hlášení viz.[8]. Potvrzení vzniklé chyby se potvrdí stiskem tlačítka OK, tím dojde k nastavení *ridici\_slovo* (9) na hodnotu *BOOL*, bit slouží k potvrzení přečtení chyby. Bez potvrzení chyby nelze pokračovat v provozu dynamometru.

## **6.4 Vlastnosti softwarových struktur**

Na software kromě kompatibility s panelem 445 byly kladeny především požadavky na bezpečnost. Aby se PLC nenacházel v nedefinovaném stavu, bylo toto ošetřeno. Pokud by došlo k přerušení spojení mezi systémem *SIMOTION* a ovládacím panelem musí to systém dynamometru rozpoznat a vyhodnotit vzniklou situaci jako chybový stav.

## **6.5 Strukturovaný text**

V dnešní době automatizační systémy kromě klasického zpětnovazebního řízení vyžadují i pokročilejší možnosti programování zaměřená na správu a řízení dat řídicích funkcí a pokročilé matematické výpočty. Pro tyto úkoly byl navržen speciální programovací jazyk strukturovaný text (ST).

Vychází z normy IEC 61131-3, která standardizuje programovací jazyky určené pro programování plc automatů, dá se říci, že ST je založena na textových strukturách této normy. ST mimo jiné zpřístupňuje uživateli tyto možnosti:

- Matematické a statické výpočty
- Správu dat
- Optimalizace procesů

Stejně jako IL patří do skupiny vyšších programovacích jazyků. Poskytuje mnoho abstraktních údajů, které popisují velmi složité funkce zjednodušeným způsobem. Oproti IL má ST následující výhody:

- Velmi komprimované formulace programových úkolů
- Jasná konstrukce programu

Protože se jedná o vyšší programovací jazyk, má i nevýhody:



- O překlad do strojového kódu se stará kompilátor, tento proces probíhá automaticky a uživatel do něj nemůže zasáhnout
- Vysoká míra abstrakce může způsobit menší efektivitu, protože komprimované programy jsou mnohdy větší a pomalejší

Ve srovnání s programovacími jazyky jako je ladder diagram a nebo FSB, nabízí ST mnohem širší možnosti

### ***Struktura***

Každý program napsaný v ST má následující základní části:

- Interface
- Implementaci
- POU (program organization unit)
- Deklaraci
- Statement section

Interface obsahuje příkazy pro import a export dat (proměnné, datové typy, funkce, funkční bloky). Pokud chceme využívat technologické balíčky je nutné je uvést do části INTERFACE. Stejně tak je zde potřeba zde nutné uvést veškeré programy, a FB, které budou použity ve výsledném programu.

Všechna klíčová slova jako názvy funkcí apod. jsou v SSC zbarvena modře. Hodnoty proměnných mají růžovou barvu a komentáře se zobrazují zeleně. Díky tomuto barevnému rozlišení je orientace v napsaném kódu jednodušší.

### ***Příkazy***

Strukturovaný text se skládá z mnoha příkazů, jednotlivé příkazy jsou odděleny středníky, na rozdíl od IL, může být na jednom řádku více příkazů anebo naopak může být jeden příkaz napsán na více řádcích. Strukturovaný text obsahuje funkce jako for, while, repeat atd.

Komentáře jsou zde vkládány většinou pomocí (\*\*). Ovšem tento formát není závazný a nemusí být dodržován

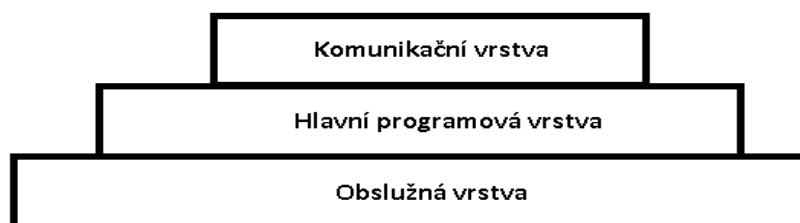
### ***Funkční bloky***

Při programování pomocí strukturovaného textu můžeme využít tzv. funkčních bloků. Systém SIMOTION Scout obsahuje mnoho předdefinovaných funkčních bloků,

také je tu možnost vytvořit si vlastní funkční blok, pro běžné aplikace jsou však dostačující již existující FB. Pokud potřebujeme vytvořit vlastní funkční blok lze využít struktury již existujících bloků V této práci je použita například struktura funkčního bloku pro usměřovač *\_ALM Control* .

## 7 Realizace software pro střídavý dynamometr

Software byl realizován pomocí strukturovaného textu. Aplikace má tři hlavní části (vrstvy). Nejvýše stojí komunikační vrstva, pod ní se nachází hlavní programová vrstva a nejnižší je obslužná vrstva. Níže uvedený obrázek znázorňuje strukturu aplikace. Komunikační vrstva je podrobněji popsána v podkapitole 7.5 V obslužné vrstvě byl použit program Gbasics, který slouží k nastavení řídicí struktury podrobnější informace jsou uvedeny v 7.8. Hlavní program slouží k nastavení struktury v Gbasics dle přijatého řídicího slova.



Obr. 7.1: struktura softwaru ovládání pohonu dynamometru

### 7.1 *Proměnné*

Mezi velmi významné pojmy v programování patří i proměnná. Jde o identifikátor, který slouží k uchovávání informací (hodnot) během spuštěného programu. V SIMOTION SCOUT rozeznáváme především dva důležité typy proměnných, a to lokální a globální proměnné. Lokální proměnné jsou využívány pouze v příslušném programu. Globální proměnné jsou viditelné v celém ST.

### 7.2 *Struktura programování*

Protože ST slouží k programování PLC, což jsou ve své podstatě cyklicky pracující stavové automaty musí tomu odpovídat i struktura programu. Programátor má na výběr mezi case a else if. V praxi se většinou kombinují obě tyto struktury. Kromě výše uvedených funkcí má ST k dispozici většinu běžně používaných programátorských funkcí jako jsou for, while atd.

Výsledný software se může skládat z jednoho jediného programu, nebo může obsahovat i více podprogramů. Díky tomu, že do programu jdou velmi snadno vkládat komentáře je i relativně dlouhý kód přehledný. Nicméně během realizace se ukázalo jako velmi užitečné oddělit datovou strukturu od samotného programu.

### **7.3 Motion Control function blocks**

Zpracováno podle [4]. Motion control function block (MCFB) slouží pro řízení pohybu pohonu. Se stále zvyšujícím se počtem aplikací, kde jsou použity PLC vznikl i požadavek na standardizaci a sjednocení nejčastěji využívaných funkcí. Právě za tímto účelem vznikly MCFB mezi jejich hlavní přednosti patří :

- Jednoduchost
- účinnost
- konzistence
- universálnost
- flexibilita

O vývoj MCFB se stará nezávislá organizace. Většina světových výrobců PLC(Siemens, Mitsubishi,...) již přijala tento nový standart a snaží se MCFB implementovat do svých softwarových vybavení.

Funkční bloky můžeme rozdělit do dvou skupin na administrativní a pohybové, podle toho, k jakému účel jsou určeny. SIMOTION obsahuje většinu těchto funkčních bloků.

### **7.4 Task skupiny**

Systém SIMOTION umožňuje námi vytvořené funkce přiřadit do různých skupin

Podle toho, v které skupině se funkce nacházejí, mají přidělenou prioritu, se kterou jsou spouštěny a rovněž se liší i procesorový čas, který je jim přidělen. Některé skupiny se vykonávají cyklicky, jiné jen při splnění určitých podmínek. Nejpoužívanější v práci byly následující skupiny:

- Start Up Task
- Motion Task

Více informací o multitaskingu je uvedeno v kapitole 3.5, Podrobnější informace jsou též uvedeny v [10].

## **7.5 Komunikační vrstva**

Na nejvyšší úrovni se nachází komunikační vrstva, slouží k navázání komunikace. Do této vrstvy patří i programy, které slouží k přijetí a odeslání dat.

Struktura přijatých a odesílaných dat je definována ve struktuře *přijatý telegram* a *odeslaný telegram*.

### **Vytvoření spojení**

První krok, který je nutné provést je vytvořit aktivní spojení mezi ovládacím panelem a systémem SIMOTION. Protože software dotykového panelu již měl plně funkční komunikaci, která jak již bylo několikrát uvedeno, využívá protokolu tcp, mohla být použita funkce *tcpOpenServer*. Systém SIMOTION je zde v pozici hostitele, zatímco ovládací dotykový panel je *slave*. Pro ověření, v jakém stavu se funkce nachází, slouží proměnná *komunikace*. Dokud ovládací panel dynamometru nenaváže spojení, zobrazuje se v levém dolním rohu dotykového panelu nápis připojování. Jestliže je spojení navázáno zobrazují se zde střídavě popisky *příjímání* a *odesílání*.

Jestliže nedojde k úspěšnému navázání komunikace, je vhodné ověřit si hodnotu proměnné parametr *function reset*.

### **Přijetí dat**

Část programu s názvem *prijem dat* se stará o přijetí telegramu z ovládacího panelu. V programu je využita funkce *tcpReceive*. Přijatá data se ukládají do proměnné *moje\_prijata\_data*. Struktura proměnné *moje\_prijata\_data* je shodná se strukturou odesílaného datového telegramu detailně popsáno v podkapitole 4.3 . Datová struktura přijatého telegramu je deklarovaná v části INTERFACE. Struktura přijatých dat je spolu se strukturou odeslaných dat pro větší přehlednost uvedena v příloze.

Při přijetí datového balíčku od ovládacího dotykového panelu je nutné zkontrolovat, že přijatá data jsou kompletní, program ověří velikost přijatých dat a

v případě že nesouhlasí jejich velikost, zahodí je a vyčká na opětovné zaslání dat. Pokud jsou přijatá data v pořádku, ovládací panel obdrží potvrzení o přijetí dat.

### ***Konverze přijatých dat***

Příchozí datový balíček je pole typu REAL o velikosti 40 bytů, ovšem například řídicí slovo musí být typu BOOL, proto je nutné typ REAL převést na typ BOOL. Řídicí slovo je v datovém balíčku reprezentováno první hodnotou typu REAL (velikost 4 byty, 32 bitů). Samotný převod probíhá v několika krocích. Nejdříve je datový typ REAL převeden na typ UDINT, z něj je následně převeden na datový typ DWORD. Poté je použita funkce pro převod z byte na bit. Funkce pro převod různých datových typů jsou uvedeny v knihovně funkcí pod položkou *conversion*. Následující obrázek znázorňuje převod přijatých dat.

Kromě výše uvedeného způsobu lze použít i funkci, která převede data přímo. Protože se ale nejedná o implicitní funkci, je nutné použít další funkci pro kontrolu převedených dat

Dalším parametrem, který je třeba převést je žádaná hodnota otáček, momentu nebo polohy, protože vstupní proměnné jsou typu LREAL. Oproti normě totiž MC funkce implementované v systému Siemens používají místo datového typu REAL LREAL.

### ***Příprava k odeslání dat***

Před samotným odesláním je nutné zjistit hodnoty všech požadovaných parametrů. Ovládací panel bude očekávat datový telegram, který se skládá z datového pole typu REAL o velikosti [0...40] bytů, požadovaná data se ale nejprve musejí převést na datový typ REAL. K tomuto účelu slouží program *priprava odeslani*.

Struktura odesílaného datového balíčku je popsána v podkapitole 4.2. Nejdůležitější pro dotykový panel je REAL s indexem [0] je zde uloženo stavové slovo, které poskytuje ovládacímu panelu informace, v jakém stavu se pohon dynamometru nachází. Naopak některé informace jako je teplota pohonu nejsou pro provoz dynamometru nezbytné a slouží panelu pouze jako dodatečný údaj o stavu pohonu.

Informace o aktuální poloze a rychlosti pohonu jsou snadno přístupné například pomocí expert listu, nebo můžeme v případě potřeby vytvořit novou proměnnou, ke které

přiřadíme příslušný parametr pomocí tečkové notace. K uvěření aktuálních hodnot rychlosti, polohy apod. je využita datová struktura v programu *GABasics*.

### ***Odesílání dat***

K odeslání byla použita funkce *tcpsend*. Jde o standardní funkci, kterou lze též použít pro komunikaci mezi systémy SIMOTION a SIMATIC. Funkce používá tyto parametry: *next command*, *datalength*, *data*. Parametr *nextcommand* určuje chování funkce. Existují dvě možné nastavení *IMMEDIATELY* a *WHEN\_COMMAND\_DONE*. Funkce vrací hodnotu typu *DINT*. Podle návratové hodnoty lze určit, zda došlo k problémům během vykonávání funkce. Existuje zde i potvrzení, že data byla úspěšně poslána.

### ***Uzavření komunikace***

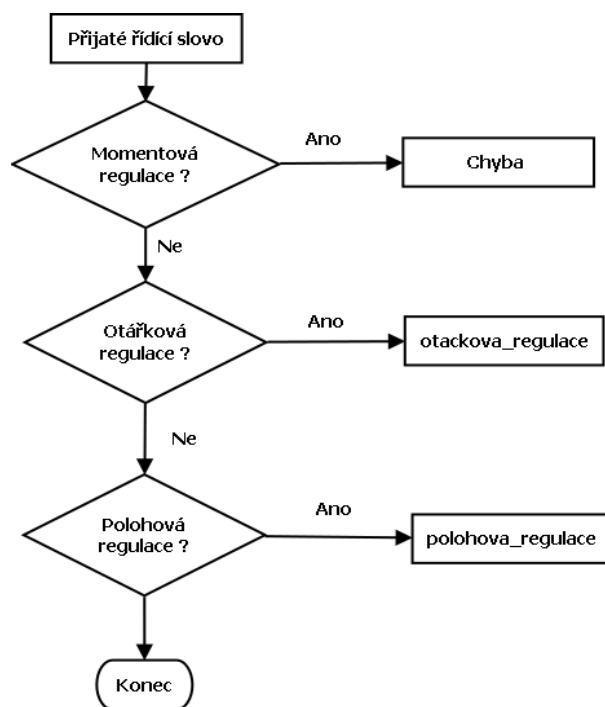
Pro uzavření komunikace je použita funkce *tcp closed*, Přesnější informace o funkci jsou uvedeny například v nápovědě programu SIMOTION SCOUT.

## **7.6 Hlavní programová vrstva**

Tato část aplikace přečte přijaté řídicí slovo a podle jeho obsahu nastaví vybrané proměnné ve spodní programové vrstvě, kde se nachází i *VGlob*al a pohon přejde do nového stavu.

### ***Volba regulace***

Část programu s názvem *hlavní program* zjišťuje, jestli uživatel zadal volbu regulace, pokud ano podle typu regulace dojde ke změně příslušných proměnných a obslužná vrstva se postará o vykonání daného typu pohybu. Navíc, pokud by uživatel zvolil momentovou regulaci, přejde motor do stavu čekání a ve stavovém slovu dojde k nastavení bitu na 1. Tento bit jak již bylo dříve uvedeno, slouží ke kontrole, zda nedošlo při provozu dynamometru k chybě. Stavové schéma je zobrazeno na obr. 12.



**Obr. 7.2: Výběr režimu regulace pohonu**

### 7.6.1 Usměrnovač

K ovládání usměrnovače byl použit funkční blok ALM Control. Pokud by došlo k přijetí řídicího slova a nebyl by zapnut ALM, a bylo vyžadováno zapnutí pohonu, k zapnutí pohonu by nedošlo a ovládací panel by měl dle formací uvedených v [1] provést nastavení:

- error :=20

Funkční blok FB ALM\_Control pouze nastaví řídicí slovo pro usměrnovač, aby se změny provedly, musí se nastavit do parametru PQW256.

Stav usměrnovače je signalizován signálkou na rozvodné skříně, je-li zapnutý, svítí i signálka. Pokud je usměrnovač zapnutý, měl by být i spuštěn jeho ventilátor. Pokud tedy uživatel vypne ventilátor usměrnovače, měl by se tím vypnout i samotný usměrnovač. Stav usměrnovače je kontrolován například při výběru regulace nebo před samotným zahájením požadovaného pohybu.

### 7.6.2 Řízení ventilátorů

Řízení ventilátorů motoru a usměrnovače provádí část programu nazvaná *řízení ventilátorů* změnou hodnoty digitálního parametru PQW 278, Podle toho jaká hodnota je do parametru zapsána jsou buďto oba ventilátory zapnuté, vypnuté, nebo běží pouze



jeden z nich. V následující tabulce jsou uvedeny všechny hodnoty digitálního parametru, spolu s předpokládaným stavem ventilátorů motoru a usměrňovače. Hodnoty jsou v hexadecimální soustavě.

**Tab. 7.1: Nastavení parametru pro ovládání ventilátorů**

Hodnota parametru	Stav ventilátoru motoru	Stav ventilátoru usměrňovače
0	Vypnuto	Vypnuto
1	Vypnuto	Zapnuto
2	Zapnuto	Vypnuto
3	Zapnuto	Zapnuto

Jak bylo uvedeno již dříve, pokud nebude zapnut ventilátor usměrňovače, nebude možné uvést pohon do chodu a dotykový panel by měl upozornit uživatele, že ventilátor je vypnutý. Pro lepší přehlednost byly digitální parametry ventilátorů přidány do watt tabulky s názvem ventilátory, díky tomu má uživatel přehled v jakém stavu se ventilátory nachází. Pokud by bylo třeba, dají se hodnoty digitálního parametru upravovat přímo přes tuto tabulku následujícím způsobem: Do pole *value* uživatel napíše požadovanou hodnotu a poté pravým tlačítkem myši klikne na tlačítko *immediatly*. Pokud by uživatel nastavil parametr na hodnotu 2, dle tabulky by se měl zapnout pouze ventilátor motoru, s praktického hlediska to není příliš vhodné.

### 7.6.3 Vyhodnocení chyb ovládacího panelu

Pokud by na pohonu došlo k chybě, kterou programátor ovládacího panelu nepředpokládal viz. 6.3, nepřihradí panel chybě příslušné číslo a uživatel tak nemůže poznat k jaké chybě došlo, tuto situaci lze vyřešit například přidáním technologického objektu pro potvrzení alarmů.

## 7.7 Datová struktura dGlobal

Nachází se zde několik datových struktur podle použití je lze rozdělit do následujících skupin:

- Příkazové
- Stavové
- Parametry
- Stav osy
- Řídící

## 7.8 *Obslužná vrstva*

Do této vrstvy patří program *gBasic\_VA*. Tento program běží cyklicky. Neustále kontroluje, zda nejsou splněny podmínky pro jednotlivé stavy pohonu. Program má definovány následující stavy:

- Chyba osy pohonu
- Chyba
- Zapnutí zdroje
- Čekání
- Homing
- Rychlostní pohyb
- Připraven
- Zastavení pohybu
- Resetován

Další důležitou funkcí je kontrola chyb, pokud nedetekuje chybu a jsou splněny podmínky, struktura přejde do určeného stavu. Na začátku programu se zjistí stav pohonu, ověří se rychlost a poloha pohonu. Zjistí se, zda se na ose nevyskytla chyba, pokud ano program přejde do stavu chyba. Dále se kontroluje, jestli je vypnuto napětí na usměrňovači, pokud ano a je potvrzena chyba na ose program přejde do stavu error reset. Jestliže se na pohonu nevyskytla chyba, program přejde do stavu čekání. Tento program má universální využití. Jsou zde použity následující MC funkce od Siemensu.

### ***Funkce MC HOME***

Slouží pro vytvoření polohové vazby mezi řízením a mechanickým systémem. Pro tento účel využívá měřicí systém. Existuje zde několik módů pro homing, pouze některé z nich lze použít pro virtuální osu.

Homing může být vyžadován pro některé MC bloky, které zpřístupňují určitý typ pohybu (MC Move Absolute) apod. V této práci byl homing použit při polohovém řízení.

### ***MC STOP***

Funkční blok ukončí všechny aktivní příkazy pro pohyb osy a zároveň zahájí zastavení pohybu osy. Průběh zastavení je dán hodnotou parametru decelerace a jerk, který je druhou derivací rychlosti, pomocí této proměnné lze zajistit, aby moment

nerostl skokově, ale narůstal postupně (momentová rampa). Po dobu, kdy je MC STOP aktivní není možné spustit žádné příkazy pro pohyb.

Funkční blok je ukončen pouze, pokud je na výstupu parametru DONE hodnota TRUE (osa stojí) a vstup Execute je resetován na hodnotu FALSE. Poté je již znovu možné použít příkazy pro pohyb osy.

### ***Funkce MC MOVE VELOCITY***

Slouží pro jednoosý pohyb požadovanou rychlostí. Kromě rychlosti blok umožňuje nastavit směr otáčení pohonu, požadovanou akceleraci, deceleraci. Podmínkou pro aktivaci je, že není aktivní MC STOP. Ukázky použití a další doplňující informace lze nalézt v [9].

**Tab. 7.2: Vstupní parametry MC MOVE VELOCITY**

Parametr	Datový typ	Počáteční hodnota	Popis
Acceleration	LREAL	-1	Parametr nastaví hodnotu zrychlení
Deceleration	LREAL	-1	Parametr nastaví hodnotu zpomalení
Velocity	LREAL		
Direction	LREAL		Nastavení požadovaného směru otáčení pohonu

### ***MC MOVE Absolute***

Funkční blok se používá pro polohové řízení osy, jak je patrné z názvu FB jedná se o polohu absolutní. Jako u většiny FB je vyžadováno, aby nebyl aktivní blok MC STOP. Nutnost HOMINGU je určena nastavením osy pohonu jestliže byl parametr *TypeOfAxis.Homing.referencing.Necessary* nastaven na ano je vyžadován homing. Následující tabulky obsahují vstupní parametry spolu s jejich popisem.

**Tab. 7.3: Vstupy MC MOVE ABSOLUTE**

Parametr	Datový typ	Počáteční hodnota	Popis
Axis	AXIS_REF	0	Specifikuje nastavení osy
Execute	BOOL	FALSE	Určuje sepnutí FB (sepnutí na náběžnou hranu signálu na vstupu)

Position	LREAL	0	Určuje absolutní polohu koncového bodu pohybu
Velocity	LREAL	-1.0	Určuje maximální rychlost, skutečná dosažená rychlost závisí na nastaveném zrychlení, jerku. Hodnoty parametrů, které lze nastavit: Hodnota > 1. Je použita nastavená hodnota. Hodnota = -1.0 je použita základní hodnota
Acceleration	LREAL	-1	Nastavení maximální hodnoty akcelerace Nastavení -1.0 znamená, že se použije hodnota parametru <i>userdefaultdynamics.positiveaccel</i> (vztahuje se vždy k ose která je použita)
Deceleration	LREAL	-1	Nastaví maximální deceleraci. Nastavení -1 použije se hodnota uvedená v parametru <i>userdefaultdynamics.negativeaccel</i>
Jerk	LREAL	-1	Nastavení maximální hodnoty jerku při nastavení -1 se použije hodnoty systémových parametrů bližší informace uvedeny v [10].
Direction	_MC_Direction	USER DEFAULT	Parametr může mít následující nastavení POSITIVE (kladný směr otáčení) NEGATIVE (záporný směr otáčení) USER_DEFAULT (původní hodnota z konfigurace osy)

FB má následující výstupní parametry: Done, Busy, Active, Command Aborted, Error, ErrorID Protože výstupní parametry nevyžadují nastavení, jsou zde uvedeny pouze jejich názvy detailnější informace lze nalézt v [11].

### ***Otáčkové řízení***

Tento režim je určen pro řízení otáček pohonu. Existují zde dva různé režimy, motorický, při němž pohon pracuje jako motor, nebo brzdový zde motor pracuje jako brzda. Otáčkové řízení využívá k pohybu MC funkci *move velocity*, Tu lze nalézt v Sscout v knihovně funkcí v části pro jednoosé operace. Více informací o funkci je uvedeno v 7.8. Maximální hodnota akcelerace a decelerace je nastavena na 7000. Nejvyšší rychlost, kterou je možno na panelu nastavit je 7500.

Pokud je vybrána otáčková regulace, hlavní program zkontroluje, zda je zadán kladný nebo záporný směr otáčení. Pokud je překročena maximální hodnota otáček, je nahrazena nejvyšší možnou hodnotou.

### ***Momentové řízení***

V případě momentového řízení je situace složitější. U otáčkového a polohového řízení SIMOTION má k dispozici MC funkce, které jsou shodné. Pro řízení momentu nemá oproti PLC open k dispozici MC funkce. Při realizaci momentového řízení bylo navýběr ze dvou variant, jak realizovat momentové řízení. První možností bylo využít rychlostní řízení spolu s omezením maximálního momentu, tato metoda sice neumožňuje uživateli nastavit momentovou rampu, ale na rozdíl od druhé varianty přímého řízení momentu je její realizace jednodušší. Druhá varianta přímého řízení momentu by byla na realizaci obtížnější, a proto bylo zvolena regulace s přesycenou rychlostní regulací.

Při realizaci byl využito kromě rychlostního řízení i parametr pro omezení maximálního momentu. Kvůli momentovému řízení musel být rozšířen komunikační telegram. Úprava parametru je realizována za využití funkce MC\_Writeparameter. A i přesto, že není možné aby si uživatel nastavil momentovou rampu je zvolené momentové řízení vyhovující.

### ***Polohové řízení***

Polohové řízení využívá především funkci MC\_moveAbsolute. Pro využití funkce je nutné, aby byl nejdříve proveden tzv. homing viz. Funkce MC\_HOME,

Po zvážení všech možností nebylo použito relativné řízení polohy, protože to by vyžadovalo dodatečné tlačítko na dotykovém panelu. Absolutní řízení polohy však reaguje na změnu zadané polohy a není proto potřeba tlačítko pro ovládání polohového řízení. Podrobnější informace o funkci MC\_move\_absolute jsou uvedeny v 7.8

### ***Kontrola stavu osy***

Aby byl provoz dynamometru bezpečný, je nutné zajistit, aby program měl přístup k informacím o stavu osy a uměl je správně a včas vyhodnotit. Je vhodné, aby kontrola stavu osy byla provedena před sekvencí pro pohyb pohonu. Pro kontrolu stavu osy byly využity standardní funkce SSC. Pokud by byla na pohonu detekována chyba, mělo by se vypnout napájení usměrňovače a tím dojít k zastavení pohonu.

### ***Změny v datové struktuře***

Protože při realizaci softwaru bylo zjištěno, že údaj o přesné teplotě usměrňovače a motoru není ve stávajícím hardwarovém zapojení dostupný, lze pouze zjistit, zda byla překročena bezpečná teplota, byl údaj o teplotě usměrňovače nahrazen údajem zobrazující číslo chyby 5. Číselná hodnota chyby byla zvolena malá, proto aby bylo patrné, že se nejedná o skutečnou teplotu.

Při testování řízení ventilátorů bylo zjištěno, že dotykový ovládací panel nesprávně zobrazuje stav ventilátorů. Reaguje pouze na stisknutí tlačítek na panelu, ale nezobrazuje skutečný stav ventilátorů, pokud je řídicí slovo změněno pomocí SScout stav ventilátorů na dotykovém panelu zůstane nezměněn. Tento problém ošetřuje část programu, nazvaná kontrola ventilátorů. Pokud dojde k vypnutí ventilátorů, při chodu pohonu či usměrňovače nastaví se bit ve stavovém slovu na TRUE, díky tomu panel detekuje chybu a pohon se zastaví.

## 8 Testování realizovaného softwaru pro úlohu dynamometru

I přesto, že systém SIMOTION Scout obsahuje nástroj pro simulaci vytvořených obslužných programů, jeho možnosti jsou však velmi omezené. Pro zjištění hodnot jednotlivých proměnných se ukázal tento integrovaný nástroj jako nevhodný, protože ne vždy byl schopen správně zobrazit požadované hodnoty. Pro ladění a testování vlastních programů je tedy vhodné využívat *watchtable*.

Jednou z prvních věcí, kterou bylo potřeba ověřit, byla funkčnost komunikace mezi dotykovým panelem a systémem SIMOTION. Pokud je úspěšně navázáno spojení ovládací panel pošle datový telegram, systém SIMOTION ho úspěšně přečte, ale neodešle zpátky požadovaná data, ovládací panel znovu otevře spojení a zkusí to znovu. Komunikace mezi ovládacím panelem a systémem Siemens je detailněji popsána v kapitole 7.5.

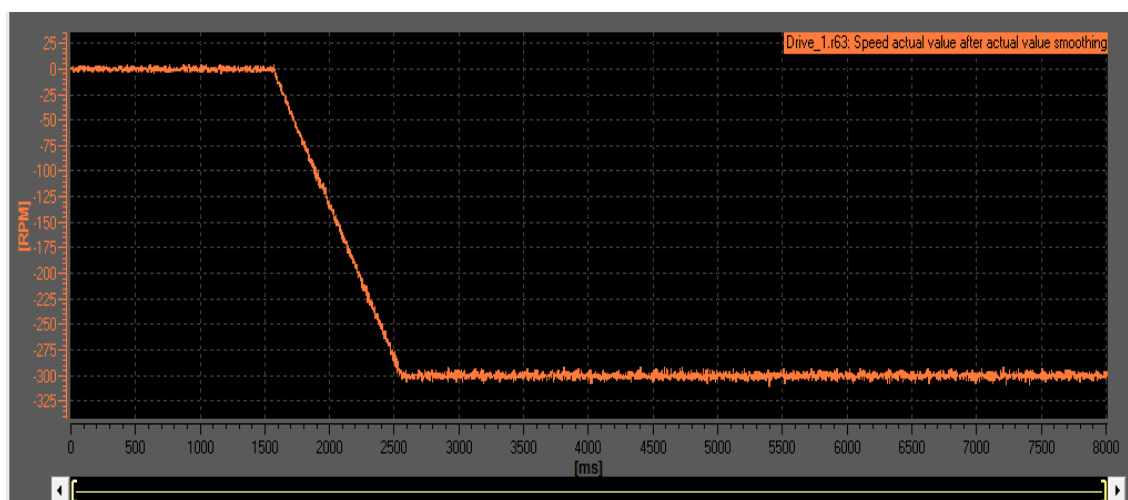
Před testováním jednotlivých módů pohybu bylo nutné potvrdit funkčnost ovládání usměrňovače a regulace ventilátorů pohonu a usměrňovače. Zde bylo zjištěno, že ovládací panel nekorektně zobrazuje skutečný stav ventilátorů a že informace uváděné v [1] nejsou zcela přesné. Naopak zobrazení stavu usměrňovače pracovalo korektně a ve shodě s informacemi uváděnými v [1].

Test blokace zapnutí pohonu při chybovém stavu dopadl úspěšně.

### ***Test otáčkového řízení***

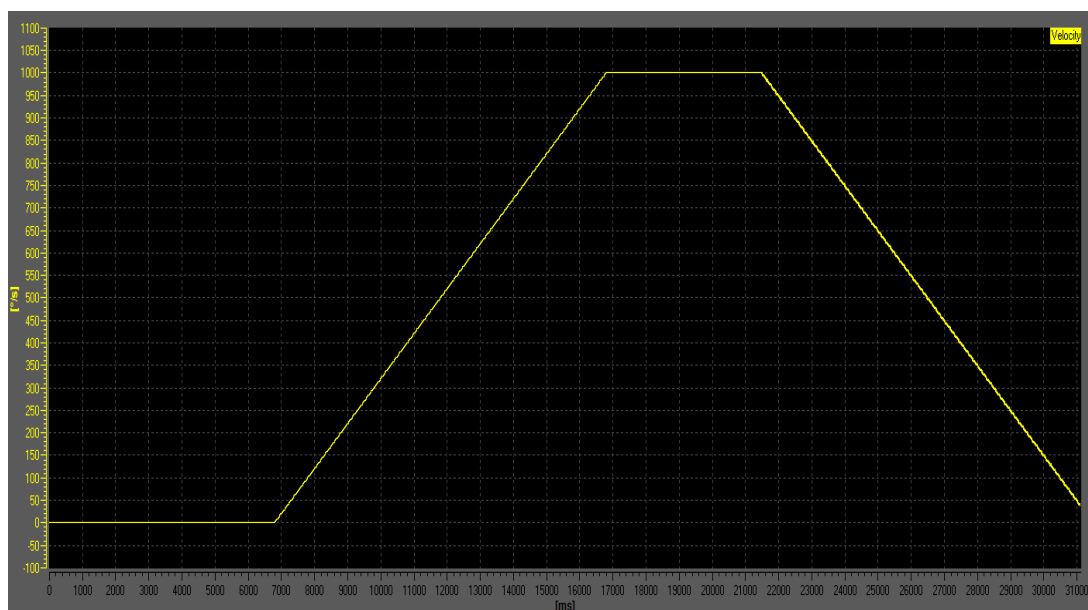
Po úspěšném ověření komunikace bylo možné přistoupit k testování otáčkového řízení. Pro následující test byly použity následující parametry: rychlost 300 ot/min. Akcelerace 300 ot/s. Směr otáčení ve směru ručiček. Následující graf ukazuje vývoj

rychlosti v čase.



**Obr.8.1: Průběh akcelerace**

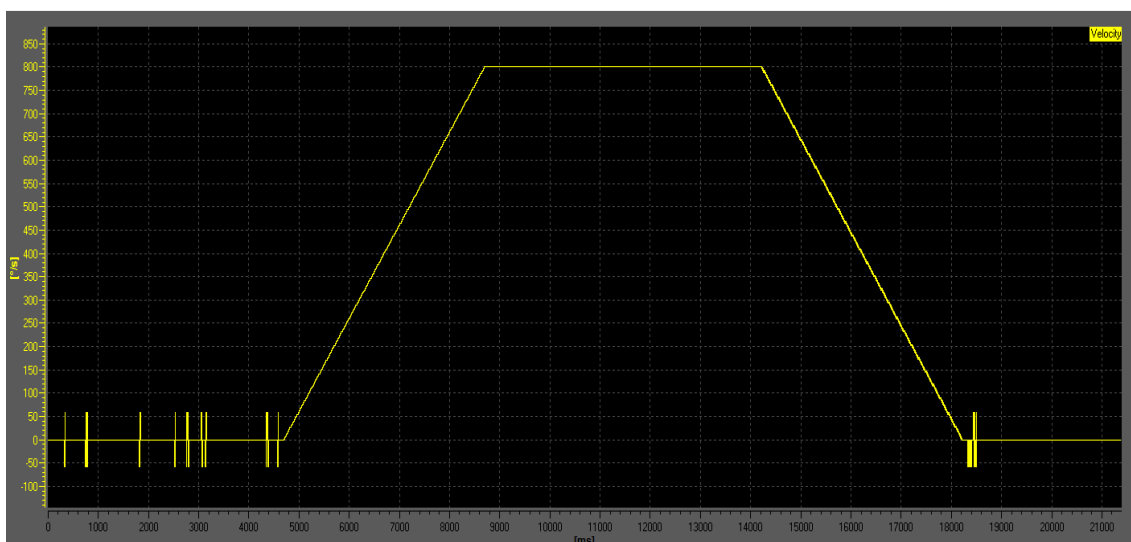
Dále byl proveden další test, parametry testu jsou následující: rychlost 1000 ot/min, akcelerace 100 ot/min, decelerace 100 ot/min.



**Obr.8.2: Rychlostní regulace test 2**

Poslední test rychlostního řízení měl parametry: žádaná rychlost 800 ot/min, akcelerace 200 ot/min, decelerace 200 ot/min.





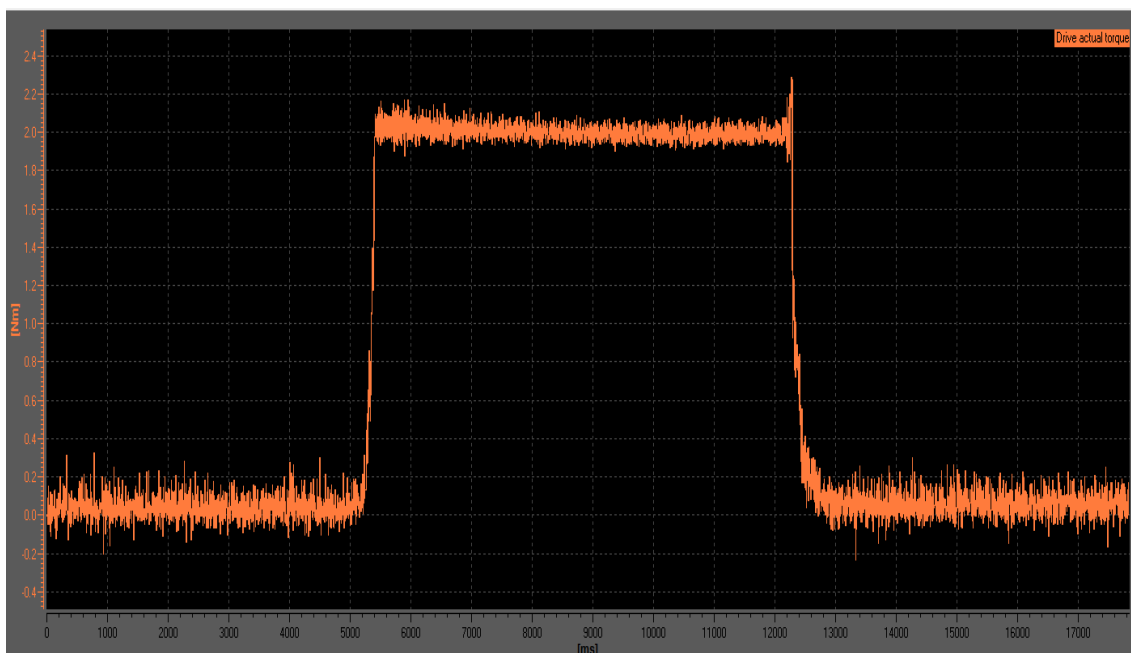
**Obr.8.3: Regulace rychlosti**

Měření bylo zaznamenáno pomocí SSC. Při otáčkovém řízení bylo úspěšně otestováno i řízení směru otáčení. Dále byla při tomto testu ověřena funkčnost vykreslování záznamu měření na ovládacím panelu. Jak je uvedeno v [1] grafické znázornění měření na dotykovém ovládacím panelu má v porovnání s měřením pomocí systému SINAMICS velmi malé rozlišení. Protože SSC obsahuje velmi dobré nástroje pro záznam měření viz. 5.1.6, je funkce grafického zobrazení měřených hodnot na ovládacím panelu dostačující pro základní měření a v případě potřeby může uživatel využít měřicích nástrojů systému SIMOTION nebo SINAMICS.

### ***Test momentového řízení***

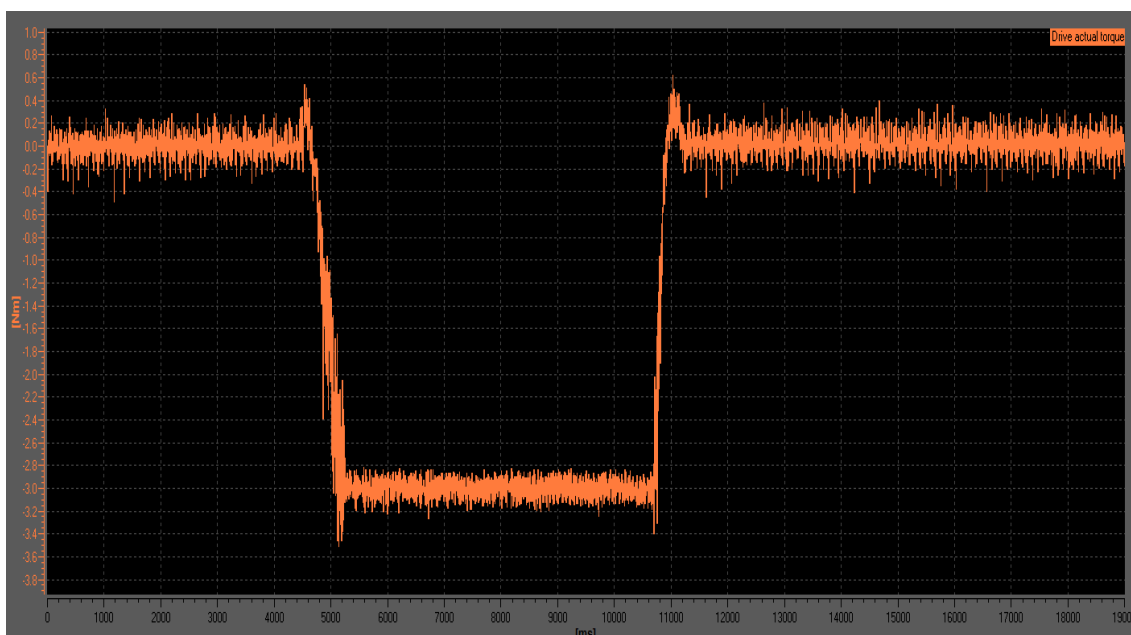
Dalším řízením, které bylo úspěšně implementováno je momentové řízení. Zde je regulovanou veličinou moment. Momentové řízení umožňuje uživateli zadat velikost výsledného momentu, naopak uživatel nemůže ovlivnit momentovou rampu, tento nedostatek není pro praktické použití důležitý.

Pro první test momentového řízení byla nastavena hodnota momentu na 2 Nm. Průběh momentu během testu je zobrazen na následujícím obrázku.



**Obr.8.4: Momentová regulace**

Pro druhý test momentového řízení byla hodnota momentu nastavena na -3 Nm. Průběh momentu je zobrazen na níže uvedeném obrázku.



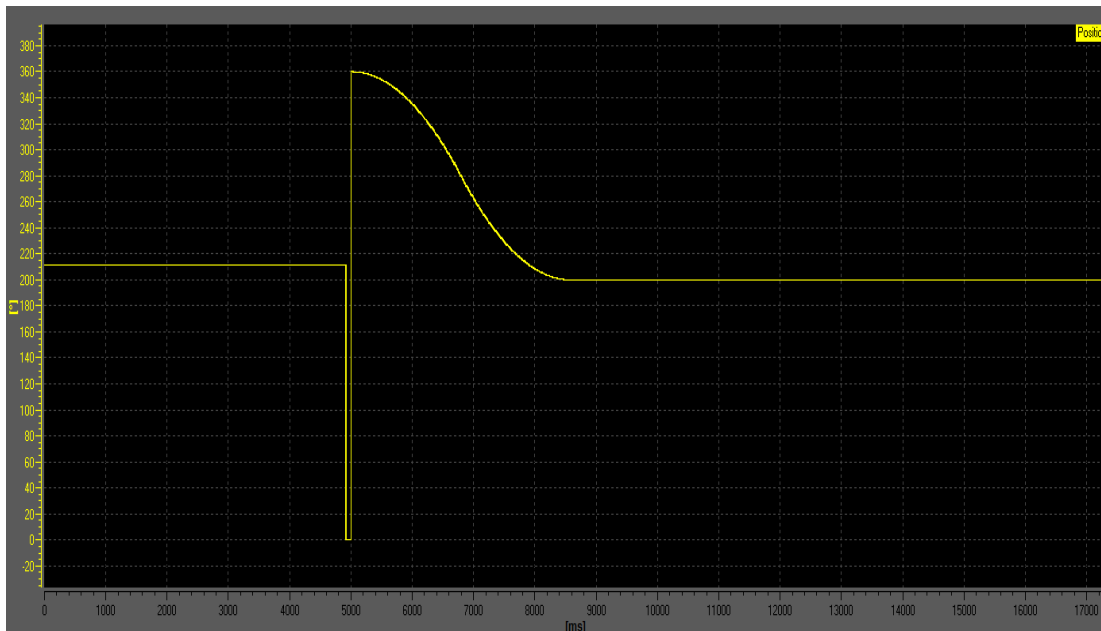
**Obr.8.5: Průběh momentu při momentové regulaci**

### ***Test polohového řízení***

Další řízení pohonu, které bylo implementováno a otestováno je polohové řízení. Bylo zvoleno absolutní polohování, tedy do pole se zadá žádaná poloha ve stupních a pohon se zastaví až při dosazení žádané polohy. Před samotným polohováním musel být

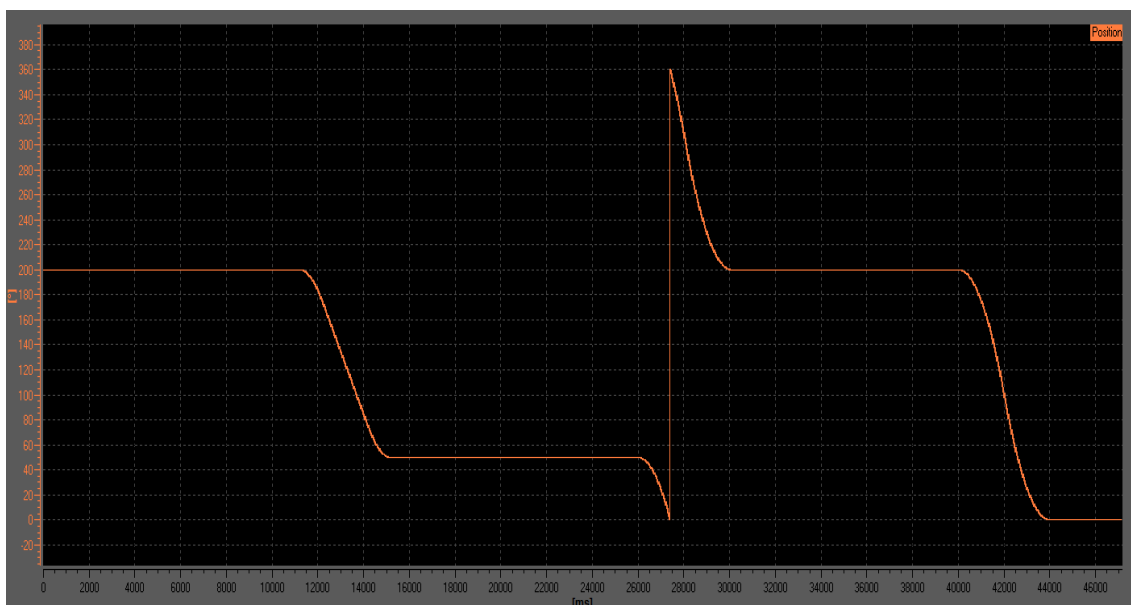
odzkoušen homing, až poté co bylo ověřeno, že je úspěšně dokončen, bylo možné přistoupit k otestování absolutní řízení polohy.

Při prvním testu byla žádaná poloha  $200^\circ$  a výchozí poloha pohonu byla  $211^\circ$  výsledek polohového řízení je zobrazen na následujícím obrázku.



**Obr.8.6: Průběh polohy při polohové regulaci test 1**

Parametry druhého testu polohového řízení byly následující: počáteční poloha  $200^\circ$  poté uživatel zadal polohu  $50^\circ$  následně  $200^\circ$  a nakonec  $0^\circ$ . Průběh testu ukazuje níže uvedený obrázek.



**Obr.8.7: Průběh polohy při polohové regulaci test 2**

## **Závěr**

Během práce jsem se seznámil se zařízením SIMOTION D425. Při tvorbě softwaru jsem se naučil programování ve vývojovém prostředí SIMOTION SCOUT, ve kterém byl software naprogramován. Zároveň jsem se seznámil s programováním PLC automatů.

K tvorbě softwarových struktur byly nutné znalosti o stavových automatech. Protože komunikace mezi panelem a systémem SIMOTION je zajištěna pomocí ethernetu a TCP/IP protokolu bylo nutné se seznámit se základními informacemi o TCP/IP protokolu

Při návrhu struktur a vlastností softwaru se vycházelo z požadavků, které vyplývaly s již existujícího softwaru panelu a TCP/IP protokolu. Pro lepší přehlednost, bylo vytvořeno několik stavových diagramů, které upřesňují, v jakých předpokládaných stavech se systém nachází

Při samotné realizaci softwaru ve vývojovém prostředí SIMOTION Scout byl použit strukturovaný text, to se ukázalo jako výhodné i přesto, že kvůli přehlednosti musel být kód okomentován. S vlastní zkušenosti mohu říci, že i přesto, že vývojové prostředí obsahuje simulátor, jeho použití má své omezení.

Software má tři způsoby řízení: otáčkovou regulaci, polohovou regulaci a momentovou regulaci. Otáčková regulace byla implementována úspěšně a je již zcela funkční. Momentová regulace byla též implementována úspěšně a je funkční. Polohová regulace je též zcela funkční.

Jako pokračování práce by bylo žádoucí upravit ovládání ventilátorů a automatizovat záznam měření, protože stávající grafické znázornění na ovládacím panelu není vyhovující.

## Seznam obrázků

Obr.2.1: Laboratorní dynamometr .....	10
Obr.2.2: Pohon dynamometru .....	11
Obr.2.3: Externí měřicí systém .....	11
Obr. 5.1: Rozhraní programu Siemens SIMOTION SCOUT .....	23
Obr. 5.2: Panel pro aktuální stav PLC D425 .....	25
Obr. 5.3: Rozhraní pro řízení vykonávání úloh ( <i>execute task system</i> ). .....	27
Obr.5.4 :Expert list.....	28
Obr. 6.1: Obrazovka setup .....	33
Obr. 6.2: nastavení dynamometru pomocí dotykového panelu .....	33
Obr. 6.3: Obrazovka ovládání pohonu dynamometru .....	34
Obr. 6.4: Průběh diagnostiky chyb ovládacího panelu .....	36
Obr. 7.1: struktura softwaru ovládání pohonu dynamometru .....	40
Obr. 7.2: Výběr režimu regulace pohonu.....	45
Obr.8.1: Průběh akcelerace .....	53
Obr.8.2: Rychlostní regulace test 2 .....	53
Obr.8.3: Regulace rychlosti .....	54
Obr.8.4: Momentová regulace .....	55
Obr.8.5: Průběh momentu při momentové regulaci.....	55
Obr.8.6: Průběh polohy při polohové regulaci test 1 .....	56
Obr.8.7: Průběh polohy při polohové regulaci test 2 .....	56

## Seznam tabulek

Tab. 4.1: Struktura odesílaných dat .....	20
Tab. 4.2: Stavové slovo pro nastavení dynamometru .....	21
Tab. 4.4: Struktura přijatého řídicího slova .....	22
Tab. 6.1: Mezní hodnoty nastavení ovládacího panelu.....	31
Tab. 7.1: Nastavení parametru pro ovládání ventilátorů.....	46
Tab. 7.2: Vstupní parametry MC MOVE VELOCITY .....	48
Tab. 7.3: Vstupy MC MOVE ABSOLUTE.....	48

## Použitá literatura

- [1]. BUREŠ, Roman. Řídicí panel dynamometru. Liberec, 2012. Bakalářská práce. Technická univerzita v Liberci, Fakulta mechatroniky, informatiky a mezioborových studií.
- [2] JOHN, Karl-Heinz; TIEGELKAMP, Michael. IEC 61131-3:programming industrial automatik systems : Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids . 2nd ed. New York : Springer, 2010. 390 s. ISBN 978-3-642-12014-5
- [3] KOVÁŘ, Josef, Zuzana PROKOPOVÁ a Ladislav ŠMEJKAL. 2008 *Programování PLC* [online]. Zlín. [cit. 2014-04-27]. Dostupné z:
- [4] PLCOPEN 2011. *Technical Committee2 – Task Force, Function blocks for motion control* [online]. Version 2.0, [cit. 2013-04-21]. Dostupné z:  
[http://www.plcopen.org/pages/tc2\\_motion\\_control/downloads/plcopen\\_mc\\_part1\\_v20.pdf](http://www.plcopen.org/pages/tc2_motion_control/downloads/plcopen_mc_part1_v20.pdf).
- [5] PUŽMANOVÁ, RITA. 2004. *TCP/IP v kostce*, 1. Vyd. České Budějovice: Kopp. ISBN 80-723-2236-2
- [6] SIEMENS. 2008. *SIMOTION D4x5* [online]. 08/2008 Edition [Cit. 2013-5-6]. Dostupné z:  
<http://www.industry.usa.siemens.com/drives/us/en/motion-control-systems-and-solutions/motion-control-products/motion-controllers/simotiond/Documents/MC-SIMOTION-D4x5-commissioning-manual.pdf>
- [7]. SIEMENS. 2010. *Simotion References* [online]. 11/2011 Edition [Cit. 2013-7-8]. Dostupné z:  
[http://cache.automation.siemens.com/dnl\\_iis/Dc/DczMDk2OQAA\\_51783855\\_HB/SIMOTION\\_References.pdf](http://cache.automation.siemens.com/dnl_iis/Dc/DczMDk2OQAA_51783855_HB/SIMOTION_References.pdf)
- [8]. SIEMENS. 2010. *D4x5 Operating* [online]. 11/2010. [Cit. 2013-10-9]. Dostupné z:  
[http://cache.automation.siemens.com/dnl/TI/TI1OTEwOQAA\\_51774900\\_HB/D4x5\\_Operating.pdf](http://cache.automation.siemens.com/dnl/TI/TI1OTEwOQAA_51774900_HB/D4x5_Operating.pdf)

- [9] SIEMENS. 2013. *SINAMICS Engineering manual* [online]. 5/2013. Dostupné z:  
<http://support.automation.siemens.com/WW/llisapi.dll/csfetch/83180185/sinamics-engineering-manual-lv-en.pdf?func=cslib.csFetch&nodeid=83180793&forcedownload=true>
- [10] SIEMENS. 2009. *SIMOTION SCOUT Configuration Manual* [online]. 5/2009. Dostupné z:  
[http://cache.automation.siemens.com/dnl/jE/jE0MjMwOQAA\\_27002825\\_HB/SCOUT\\_en-US.pdf](http://cache.automation.siemens.com/dnl/jE/jE0MjMwOQAA_27002825_HB/SCOUT_en-US.pdf)
- [11] SIEMENS. 2008, *PLCopen Blocks Function Manual* [online]. 08/2008 Dostupné z:  
[http://cache.automation.siemens.com/dnl\\_iis/Tc/TcyNzgyNwAA\\_27002442\\_HB/PLCopen\\_Functions\\_en-en-US.pdf](http://cache.automation.siemens.com/dnl_iis/Tc/TcyNzgyNwAA_27002442_HB/PLCopen_Functions_en-en-US.pdf)

## ***Příloha A***

### ***D425 Technické informace***

Data		Data	
Maximální počet os	16	Maximální počet os s využitím integrovaného řízení (servo/vector/V/f)	(6/4/8 )
Minimální čas PROFIBUS cyklu	2 ms	Počet digitálních vstupů	8
Napájecí napětí	24 VDC	Zapínací proud	6A
RAM	17 MB	Diagnostický buffer	200 zpráv
Digitální vstupy			
Maximální počet		8	
Použito jako vstup			
Vstupní napětí		24 VDC	
Vstupní napětí se signálem "1"		15..30	
Vstupní napětí se signálem "0"		-3 až +5 V	
Galvanicky odděleno		ne	



## ***Příloha B***

### **Seznam přiloženého cd**

- 1. Diplomová práce**
- 2. Dokumentace**
- 3. Program**